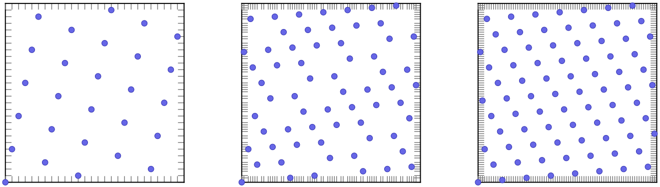
High-Dimensional Integration

Dirk Nuyens — NUMA, KU Leuven, Belgium



CIME Summer School on High-Dimensional Approximation
Cetraro, Italy

September 2024

The promises

- High-dimensional integration by
- Quasi-Monte Carlo (QMC) methods and
- how to beat “the curse of dimensionality”.

The promises

- High-dimensional integration by
- Quasi-Monte Carlo (QMC) methods and
- how to beat "the curse of dimensionality".

But, when can you beat the curse?

The promises

- High-dimensional integration by
- Quasi-Monte Carlo (QMC) methods and
- how to beat "the curse of dimensionality".

But, when can you beat the curse?

Show me the fine print...

The promises

- High-dimensional integration by
- Quasi-Monte Carlo (QMC) methods and
- how to beat "the curse of dimensionality".

But, when can you beat the curse?

Show me the fine print...

In three lectures

1. Introduction to high-dimensional integration
2. Weighted function spaces and tractability
3. Advanced topics

Topics

Topics:

- Weighted Sobolev spaces of mixed smoothness
- Periodic setting: Lattice rules
- Non-periodic setting: Interlaced polynomial lattice rules
- Dimension-independent bounds (tractability)

I will try to divide the material in

- "easy",
- "intermediate" and
- "expert" levels.

Lecture 1:
Introduction to high-dimensional integration

The plan for today

- A light introduction to "lattice points" & "lattice rules".
- Usage for numerical integration of "periodic" functions.
- Analysis of the error.
- Some words on function spaces and the worst-case error.
- Some Julia code to demonstrate things...

1. One-dimensional integrals

One-dimensional integrals

Suppose you want to integrate

$$\int_0^2 \exp(-x^5 + x)\,\mathrm{d}x.$$

What do *you* do?

One-dimensional integrals

Suppose you want to integrate

$$\int_0^2 \exp(-x^5 + x)\,\mathrm{d}x.$$

What do *you* do? No closed form...

One-dimensional integrals

Suppose you want to integrate

$$\int_0^2 \exp(-x^5 + x)\,\mathrm{d}x.$$

What do *you* do? No closed form...

Quadratures:

- Gaussian quadrature
- Newton–Cotes
- Simpson rule
- Trapezoid rule
- ...

One-dimensional integrals

Suppose you want to integrate

$$\int_0^2 \exp(-x^5 + x)\,\mathrm{d}x.$$

What do *you* do? No closed form...

Quadratures:

- Gaussian quadrature
- Newton–Cotes
- Simpson rule
- Trapezoid rule
- ...

Automatic integration? (E.g. quadpack.)

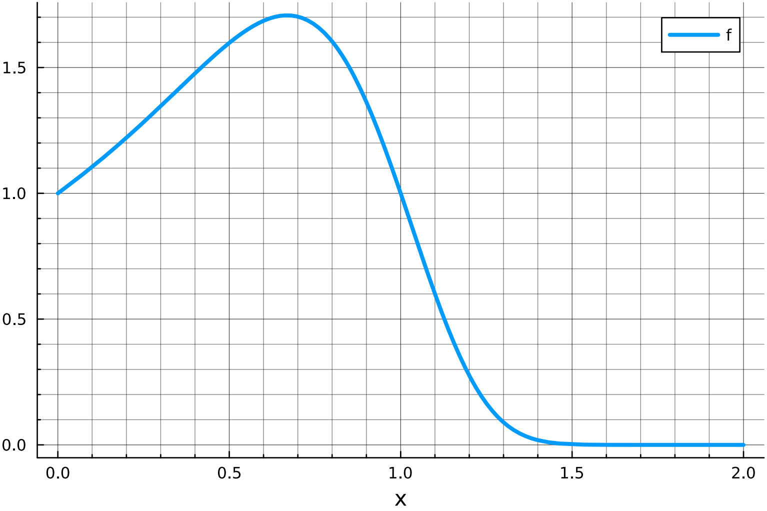
Excursion in Julia

Let's try some experiment.
I will do quite specific things.
They will generalise to high dimensions later.

(The code on the slides is for reference. I try to be complete, but I will be executing it live.)

```
f = x -> exp(-x^5 + x)
domain = (0, 2)

true_value = 1.5596783055678136 # up to double precision
```

Our client over $[0, 2]$



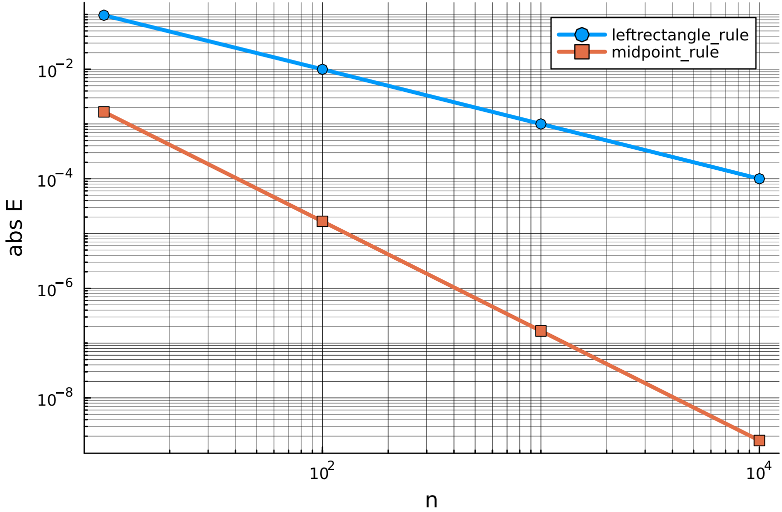
Mapping to $[0, 1]$

```
# Map point x in [0, 1] to given range [a, b], assuming a < b.
map_range(a, b) = x -> x * (b - a) + a
vol_range(a, b) = prod(b .- a)

# Convenience functions:
map_range(ab) = map_range(ab[1], ab[2])
vol_range(ab) = vol_range(ab[1], ab[2])
```

Some easy quadrature rules: rect and midp

```
n = 10

# Leftrectangle rule
xs = map_range(domain).((0:n-1)/n)
sum(f.(xs)) / n * vol_range(domain)

# Midpoint rule
xs = map_range(domain).((0:n-1)/n .+ 1/(2*n))
sum(f.(xs)) / n * vol_range(domain)

# as functions (o needs to be \circ for function composition):
leftrectangle_rule(f, domain, n) =
 sum(f o map_range(domain),
     (0:n-1)/n) / n * vol_range(domain)
midpoint_rule(f, domain, n) =
 sum(f o map_range(domain),
     (0:n-1)/n .+ 1/(2*n)) / n * vol_range(domain)
```

Convergence as expected. . .



What about trapezoid rule?

What about trapezoid rule?

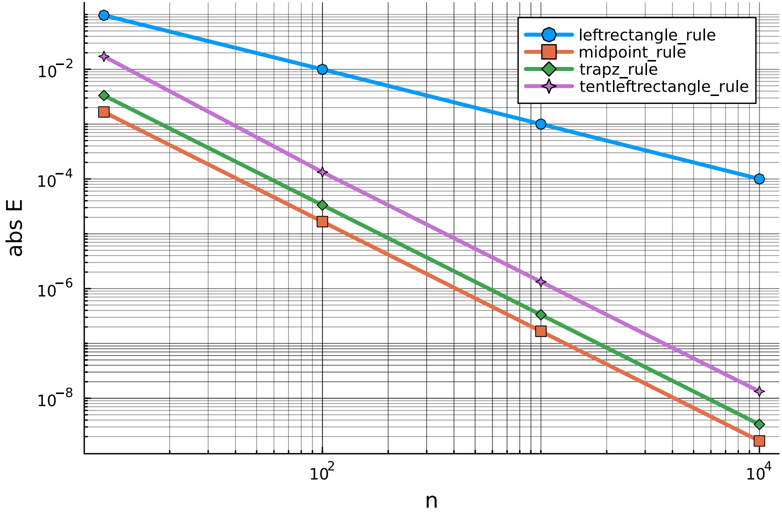
Glad you asked!

- Trapezoid should also give n^{-2} convergence.
- Advantage of trapezoid over midpoint?
  You can keep on doubling the number of points...
- Very important: left rectangle and trapezoid rules are like taking simple averages: the weights change in an obvious way if you double the number of points!

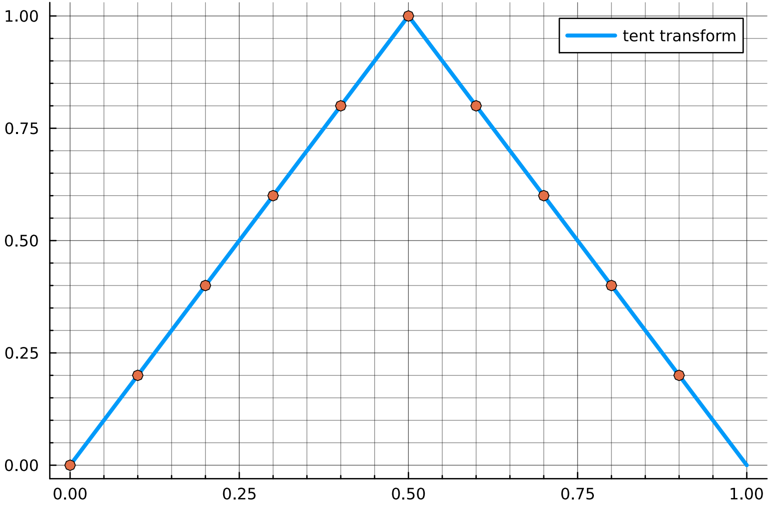
Trick: tent-transform. Lebesgue preserving.

```
tent(x) = 1 .- abs.(1 .- 2*x)
```

Convergence as expected...



Tent transform



Tent transform

The trapezoid rule can be obtained by using the tent-transform:

```
sum(f o map_range(domain) o tent,
    (0:n-1)/n) / n * vol_range(domain)
```

We will use this in higher dimensions as well...

What do *you* remember about error bounds?

Tabel 8.2: Samengestelde kwadratuurformules van graad n met m deelintervallen

$$h = \begin{cases} (b-a)/m & \text{voor } n = 0, \\ (b-a)/(nm) & \text{voor } n \geq 1, \end{cases} \qquad N = \begin{cases} m & \text{voor } n = 0, \\ nm+1 & \text{voor } n \geq 1, \end{cases}$$

$$x_i = a + i\,h, \qquad i = 0, \ldots, N-1 \qquad (\text{uitz: } x_i^* = a + i\,h + h/2 \text{ voor } Q_{0,m}^*)$$

graad	regel $Q_{n,m}$	grens voor $\lvert R_{n,m}\rvert$
$n=0$	$Q_{0,m}(f) = h \sum_{i=0}^{N-1} f(a+i\,h)$	$\lvert R_{0,m}\rvert \leq \frac{1}{2}\,\lVert f'\rVert_\infty \frac{(b-a)^2}{N}$
	$Q_{0,m}^*(f) = h \sum_{i=0}^{N-1} f(a+i\,h+h/2)$	$\lvert R_{0,m}^*\rvert \leq \frac{1}{24}\,\lVert f''\rVert_\infty \frac{(b-a)^3}{N^2}$
$n=1$	$Q_{1,m}(f) = h\,\frac{f(a)+f(b)}{2} + h \sum_{i=1}^{N-2} f(a+i\,h)$	$\lvert R_{1,m}\rvert \leq \frac{1}{12}\,\lVert f''\rVert_\infty \frac{(b-a)^3}{(N-1)^2}$
$n=2$	$Q_{2,m}(f) = h\,\frac{f(a)+f(b)}{3} + h \sum_{i=1,3,\ldots,N-2} \frac{4}{3}\,f(a+i\,h) + h \sum_{i=2,4,\ldots,N-3} \frac{2}{3}\,f(a+i\,h)$	$\lvert R_{2,m}\rvert \leq \frac{1}{180}\,\lVert f^{(4)}\rVert_\infty \frac{(b-a)^5}{(N-1)^4}$

What do *you* remember about error bounds?

We have indeed that

- Left rectangle rule is n^{-1} for "smoothness 1".
- Midpoint rule is n^{-2} for "smoothness 2".
- Trapezoid is n^{-2} for "smoothness 2".

Smoothness is in terms of derivatives: Sobolev spaces.

But: for periodic functions the trapezoidal rule can give $n^{-\alpha}$ for $\alpha > 2$ and even exponential convergence... Why is that? Next...

N.B. for a periodic function $f(0) = f(1)$ and the left rectangle rule is identical to the trapezoidal rule.

Also for non-periodic functions a transformed trapezoidal rule can give higher order convergence: Clenshaw–Curtis rules...

"Periodic" functions over $\mathbb{T} \simeq [0, 1)$

Suppose

$$f(x) = \sum_{h \in \mathbb{Z}} \hat{f}(h) \exp(2\pi \mathrm{i}\, hx)$$

with

$$\hat{f}(h) = \int_0^1 f(x) \exp(-2\pi \mathrm{i}\, hx) \,\mathrm{d}x$$

and absolutely summable Fourier coefficients

$$\sum_{h \in \mathbb{Z}} |\hat{f}(h)| < \infty.$$

We have

$$I(f) := \int_0^1 f(x) \,\mathrm{d}x = \hat{f}(0).$$

Quadrature error for periodic function

We have for the trapezoidal rule (or left rectangle rule):

$$\begin{aligned}
Q_n^T(f) - I(f) &= \frac{1}{n} \sum_{k \in \mathbb{Z}_n} f(k/n) - \int_0^1 f(x)\,\mathrm{d}x \\
&= \frac{1}{n} \sum_{k \in \mathbb{Z}_n} f(k/n) - \hat{f}(0) \\
&= \sum_{0 \neq h \in \mathbb{Z}} \hat{f}(h) \left[\frac{1}{n} \sum_{k \in \mathbb{Z}_n} \exp(2\pi \mathrm{i}\, hk/n) \right] \\
&= \sum_{\substack{0 \neq h \in \mathbb{Z} \\ h \equiv 0 \pmod{n}}} \hat{f}(h).
\end{aligned}$$

Only Fourier frequencies that are multiples of n contribute to error.

Quadrature error bound for periodic function

Thus, for $\alpha > 1/2$,

$$
\begin{aligned}
|Q_n^T(f) - I(f)| &= \Big| \sum_{\substack{0 \neq h \in \mathbb{Z} \\ h \equiv 0 \pmod n}} \hat{f}(h) \Big| \\
&= \Big| \sum_{\substack{0 \neq h \in \mathbb{Z} \\ h \equiv 0 \pmod n}} \hat{f}(h) \, \frac{|2\pi h|^\alpha}{|2\pi h|^\alpha} \Big| \\
&\le \Big(\sum_{0 \neq h \in \mathbb{Z}} |\hat{f}(h)|^2 \, |2\pi h|^{2\alpha} \Big)^{1/2} \Big(\sum_{\substack{0 \neq h \in \mathbb{Z} \\ h \equiv 0 \pmod n}} |2\pi h|^{-2\alpha} \Big)^{1/2} \\
&= \|f^{(\alpha)}\|_{L_2} \, \frac{\sqrt{2\zeta(2\alpha)}}{(2\pi n)^\alpha}.
\end{aligned}
$$

So if $f^{(\tau)}$ is periodic, i.e., $f^{(\tau)}(0) = f^{(\tau)}(1)$, for $\tau = 0, \ldots, \alpha - 1$ and $f^{(\alpha)} \in L_2$, then we have convergence $n^{-\alpha}$.

2. Higher dimensions

Discussion: what about higher dimensions?

Do *you* think the following integral is hard?

$$\int_{[0,2]^d} \prod_{j=1}^{d} \exp(-x_j^5 + x_j)\,\mathrm{d}\boldsymbol{x}$$

This is just the tensor product of our previous function.

Discussion: what about higher dimensions?

Do *you* think the following integral is hard?

$$\int_{[0,2]^d} \prod_{j=1}^{d} \exp(-x_j^5 + x_j)\,\mathrm{d}\boldsymbol{x}$$

This is just the tensor product of our previous function.
Assume black box, otherwise yes, just do $\prod_{j=1}^{d} \int_0^2 \exp(-x_j^5 + x_j)\,\mathrm{d}x_j$.

Discussion: what about higher dimensions?

Do *you* think the following integral is hard?

$$\int_{[0,2]^d} \prod_{j=1}^{d} \exp(-x_j^5 + x_j) \, \mathrm{d}\boldsymbol{x}$$

This is just the tensor product of our previous function.
Assume black box, otherwise yes, just do $\prod_{j=1}^{d} \int_0^2 \exp(-x_j^5 + x_j) \, \mathrm{d}x_j$.

- The volume is 2^d. The absolute error will scale with the volume.
- The difference between larger than 1 and smaller than 1 values from the one-dimensional function will blow up exponentially for the d-dimensional function.
- Looks like an innocent product, but, even,

$$\prod_{j=1}^{d}(1 + x_j) = \sum_{\mathfrak{u} \subseteq \{1,\ldots,d\}} \Big(\prod_{j \notin \mathfrak{u}} 1\Big)\Big(\prod_{j \in \mathfrak{u}} x_j\Big) = \sum_{\mathfrak{u} \subseteq \{1,\ldots,d\}} \prod_{j \in \mathfrak{u}} x_j$$

and $\exp(x_j) = 1 + x_j - x_j^2/2 + \cdots$. (Remember this $\uparrow$ formula.)

How to generalise quadrature formulae to higher dimensions?

Aim:

- Uncomplicated.
- No "curse by construction".
- Optimal convergence.
- Possibility for dimension-independent bounds.

What about product rules?

$$\sum_{k_1=1}^{n_1} w_{k_1}^1 \cdots \sum_{k_d=1}^{n_d} w_{k_d}^d \, f(x_{k_1}^1, \ldots, x_{k_d}^d)$$

E.g. take product of $n_j = 10$ point rule in $d = 10$ dimensions.
How many points is that?

How to generalise quadrature formulae to higher dimensions?

Aim:

- Uncomplicated.
- No "curse by construction".
- Optimal convergence.
- Possibility for dimension-independent bounds.

What about product rules?

$$\sum_{k_1=1}^{n_1} w_{k_1}^1 \cdots \sum_{k_d=1}^{n_d} w_{k_d}^d \, f(x_{k_1}^1, \ldots, x_{k_d}^d)$$

E.g. take product of $n_j = 10$ point rule in $d = 10$ dimensions.
How many points is that? What about $d = 100$?

How to generalise quadrature formulae to higher dimensions?

Aim:

- Uncomplicated.
- No "curse by construction".
- Optimal convergence.
- Possibility for dimension-independent bounds.

What about product rules?

$$\sum_{k_1=1}^{n_1} w_{k_1}^1 \cdots \sum_{k_d=1}^{n_d} w_{k_d}^d \, f(x_{k_1}^1, \ldots, x_{k_d}^d)$$

E.g. take product of $n_j = 10$ point rule in $d = 10$ dimensions.
How many points is that? What about $d = 100$?
Ok we could have decreasing n_j, but then something like
$n_{11} = n_{12} = n_{13} = \cdots = 1$ must happen. Good intuition though.
So the question is: how high dimensional?

THE CURSE OF DIMENSION!

Higher dimension generalisations

So all depends on how high dimensional.

- Sparse grid and Smolyak constructions: but still scales like $n\,(\log n)^{d-1}$ points.
- Monte Carlo sampling: but only $n^{-1/2}$ convergence.
- Quasi-Monte Carlo sampling.

We are going for something that works for really high dimensions.
(Yes I mean Monte Carlo and quasi-Monte Carlo.)

3. Lattice rules

Lattice rule = equal weight cubature using lattice points

For $f \in \mathcal{H}_\alpha$ approximate the d-dimensional integral

$$I(f) := \int_{[0,1]^d} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$$

by an n-point lattice rule with generating vector $\boldsymbol{z} \in \mathbb{Z}_n^d$

$$Q_{n,\boldsymbol{z}}(f) := \frac{1}{n} \sum_{k \in \mathbb{Z}_n} f\left(\frac{\boldsymbol{z} k \bmod n}{n} \right).$$

Worst-case error for $f \in \mathcal{H}_\alpha$ for a given algorithm Q_n (e.g. $Q_{n,\boldsymbol{z}}$):

$$e^{\mathrm{det}}(Q_n, \mathcal{H}_\alpha) := \sup_{\substack{f \in \mathcal{H}_\alpha \\ \|f\|_\alpha \leq 1}} |I(f) - Q_n(f)|.$$

Lattice rule = equal weight quadrature using lattice points

For $f \in \mathcal{H}_\alpha$ approximate the d-dimensional integral

$$I(f) := \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

by an n-point lattice rule with generating vector $\boldsymbol{z} \in \mathbb{Z}_n^d$

$$Q_{n,\boldsymbol{z}}(f) := \frac{1}{n} \sum_{k \in \mathbb{Z}_n} f\left(\frac{\boldsymbol{z}k \bmod n}{n}\right).$$

Worst-case error for $f \in \mathcal{H}_\alpha$ for a given algorithm Q_n (e.g. $Q_{n,\boldsymbol{z}}$):

$$e^{\mathrm{det}}(Q_n, \mathcal{H}_\alpha) := \sup_{\substack{f \in \mathcal{H}_\alpha \\ \|f\|_\alpha \le 1}} |I(f) - Q_n(f)|.$$

Lattice rule = equal weight cubature using lattice points

For $f \in \mathcal{H}_\alpha$ approximate the d-dimensional integral

$$I(f) := \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

by an n-point lattice rule with generating vector $\boldsymbol{z} \in \mathbb{Z}_n^d$

$$Q_{n,\boldsymbol{z}}(f) := \frac{1}{n} \sum_{k \in \mathbb{Z}_n} f\left(\frac{\boldsymbol{z}k \bmod n}{n}\right).$$

Worst-case error for $f \in \mathcal{H}_\alpha$ for a given algorithm Q_n (e.g. $Q_{n,\boldsymbol{z}}$):

$$e^{\det}(Q_n, \mathcal{H}_\alpha) := \sup_{\substack{f \in \mathcal{H}_\alpha \\ \|f\|_\alpha \le 1}} |I(f) - Q_n(f)|.$$

$\rightsquigarrow$ For good lattice rule $Q_{n,\boldsymbol{z}}$ converges like $n^{-\alpha} \|f\|_\alpha$. Optimal (Bakhvalov '59): matching upper and lower bounds (modulo logs).

Julia – Simple lattice rule example

Given n and $\boldsymbol{z} \in \mathbb{Z}_n^d$:

$$\boldsymbol{x}_k := \frac{k\boldsymbol{z} \bmod n}{n}, \qquad Q_{n,\boldsymbol{z}}(f) := \frac{1}{n} \sum_{k \in \mathbb{Z}_n} f(\boldsymbol{x}_k).$$

```
lattice_points(z, n) = ( ( (k * z) .% n ) ./ n for k in 0:n-1 )

# example function
f = r -> x -> abs(sum(exp.(2*pi*im*x)))^r

z = [1, 55]; n = 89  # Fibonacci lattice rule
mean(x -> f(1, x), lattice_points(z, n))
```

Old slide: Matlab & Python

One-liners in Matlab/Octave, Python, ...

So assume you are given a "good" $\boldsymbol{z}$ for your choice of N, then

$$Q_N(f;\boldsymbol{z},N) = \frac{1}{N}\sum_{k=0}^{N-1} f\left(\frac{\boldsymbol{z}k \bmod N}{N}\right), \quad \text{and take eg } f(\boldsymbol{x}) = \left|\sum_{j=1}^{d} e^{2\pi i x_j}\right|^r.$$

(Example $I(f)$ is expected distance of rth moment of distance travelled by d-step random walk in the plane, see, Borwein, N., Straub, Wan (2011).)

```
z = [1; 55]; N = 89; % points as [d x N] (Fortran), dim=1,2
f = @(r, x) abs( sum( exp(2*pi*1i*x), 1 ) ).^r;
% one-liner:
Q = mean( f(1, mod(z*(0:N-1), N)/N ) )
```

```
from numpy import *; # using Numpy
z = [1, 55]; N = 89; # points as [N x d] (C), axis=0,1
f = lambda r, x: abs( sum( exp(2*pi*1j*x), axis=1 ) )**r
# one-liner:
Q = mean( f(1, (outer(range(N), z) % N)/float(N) ) )
```

Deterministic vs randomized lattice rules

Deterministic worst-case error for $f \in \mathcal{H}_\alpha$ for a given algorithm Q_n:

$$e^{\mathrm{det}}(Q_n, \mathcal{H}_\alpha) := \sup_{\substack{f \in \mathcal{H}_\alpha \\ \|f\|_\alpha \le 1}} |I(f) - Q_n(f)|.$$

$\rightsquigarrow$ For good lattice rule $Q_{n,\boldsymbol{z}}$ converges like $n^{-\alpha} \|f\|_\alpha$. Optimal (Bakhvalov '59): matching upper and lower bounds (modulo logs).

For a random family of deterministic rules $Q_n^* := \{Q_n^\omega\}_\omega$:

Randomized error or worst-case expected error for $f \in \mathcal{H}_\alpha$:

$$e^{\mathrm{ran}}(Q_n^*, \mathcal{H}_\alpha) := \sup_{\substack{f \in \mathcal{H}_\alpha \\ \|f\|_\alpha \le 1}} \mathbb{E}_\omega[\,|I(f) - Q_n^\omega(f)|\,].$$

$\rightsquigarrow$ Possible to get $n^{-\alpha-1/2} \|f\|_\alpha$. (Optimal. Bakhvalov.)
(For lattice rules: randomize number of points.
Kritzer, Kuo, Nuyens, M. Ullrich (2019), ...)

Error estimation by randomization

Easy way to randomize a lattice rule is by a random shift.
Given a shift $\boldsymbol{\Delta} \in [0,1)^d$ the kth shifted point becomes

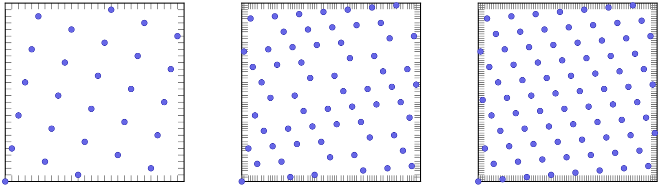
$$\boldsymbol{x}_k(\boldsymbol{\Delta}) = \left(\frac{k\boldsymbol{z} \bmod n}{n} + \boldsymbol{\Delta}\right) \bmod 1$$
$$= \left(\frac{k\boldsymbol{z}}{n} + \boldsymbol{\Delta}\right) \bmod 1 \qquad = \qquad \left\{\frac{k\boldsymbol{z}}{n} + \boldsymbol{\Delta}\right\}.$$

A shifted lattice rule:

$$Q_{n,\boldsymbol{z}}(f;\boldsymbol{\Delta}) = \frac{1}{n} \sum_{k \in \mathbb{Z}_n} f(\boldsymbol{x}_k(\boldsymbol{\Delta})).$$

If you take M shifts i.i.d. $U([0,1)^d)$ then you have M independent observations $\{Q_{n,\boldsymbol{z}}(f;\boldsymbol{\Delta}_i)\}_{i=1}^M$.
$\Rightarrow$ Calculate mean and standard error. (Construct CI.)

Lattice sequences



Normally in base 2.

Can use the van der Corput sequence for enumerating.

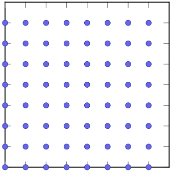
Or if only interested in totals of 2^m: evaluate the next odd indices.

"Monte Carlo type" methods: $\frac{1}{n}\sum_{k=1}^{n} f(\boldsymbol{x}_k)$

What kind of cubature/quadrature method to use for d large?

- A product of classical quadrature rules? (Product of weights!)
  $\rightarrow n = m^d \Rightarrow$ The curse "by construction"!
- The plain Monte Carlo method: $\boldsymbol{x}_k \sim U[0,1)^d$.
  $\rightarrow$ Free to choose n.
- Quasi-Monte Carlo methods: using some algebraic structure.
  $\rightarrow$ Free to choose n.

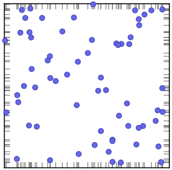
grid



$n = m^d$

error $= O(n^{-r/d})$

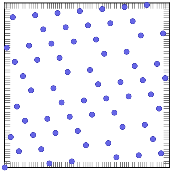
MC



n free

std $= O(n^{-1/2})$

QMC



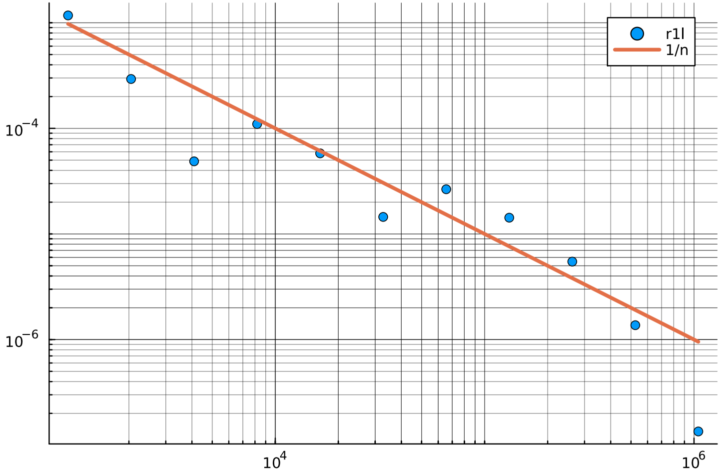
n free

error $= O(n^{-1})$, ...

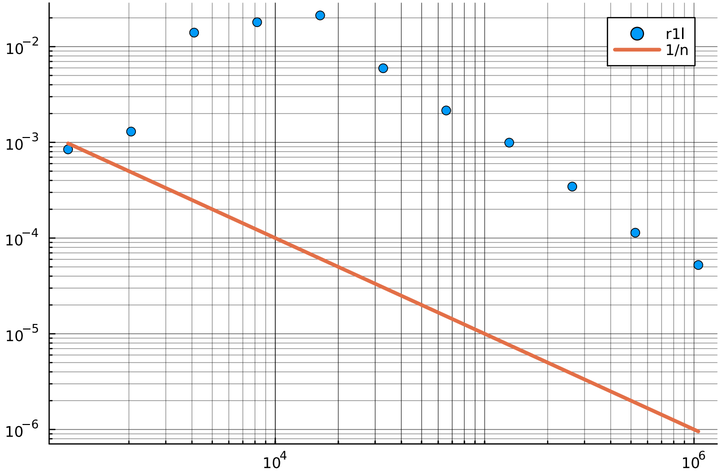
Julia – Lattice sequence in base 2 (as a plain rule sequence)

```
# exew_base2_m20_a3_HKKN.txt from Magic Point Shop:
z = [1, 364981, 245389, 97823, 488939, 62609, 400749, 385317,
     21281, 223487]       # 10 dimension with max 2^20 points

lr_seq(d, z, m1, m2) =
  ( lattice_points(z[1:d], 2^m) for m in m1:m2 )

d = 2; m1 = 10; m2 = 20;

# Simple test function which integrates to 1
f = x -> prod(1 .+ (x .- 1/2))

# Such nice vectorisation...
Es = abs.(mean.(f, lr_seq(d, z, m1, m2)) .- 1) # true integral=1

ns = 2 .^ (m1:m2)
scatter(ns, Es, xscale=:log10, yscale=:log10)
plot!(ns, ns .^ -1, xscale=:log10, yscale=:log10)
```

Absolute error versus n for $d = 2 \to$ order 1 convergence



Absolute error versus n for $d = 10 \to$ order 1 after bump



What do we see?

- The curse of dimensionality...
- Why does this happen?
  I promised you no curse...
  It depends on the function space...
- When does this happen?
  Or when does this not happen... next...
- And how do you know you have a good generating vector $\boldsymbol{z}$?

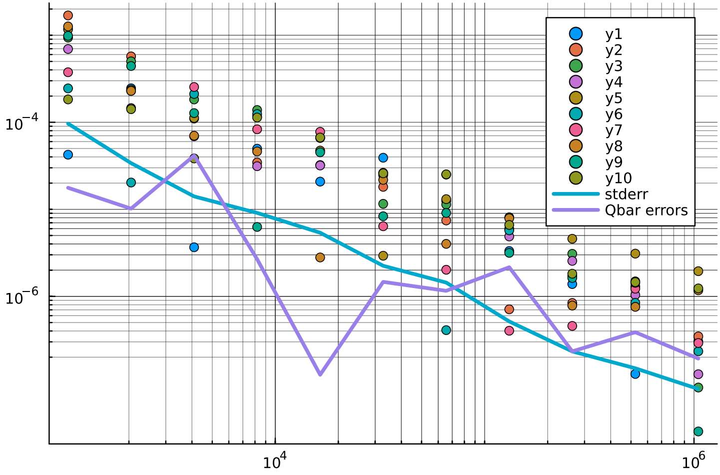
All big questions which we will try to answer tomorrow.

One last example: random shifting

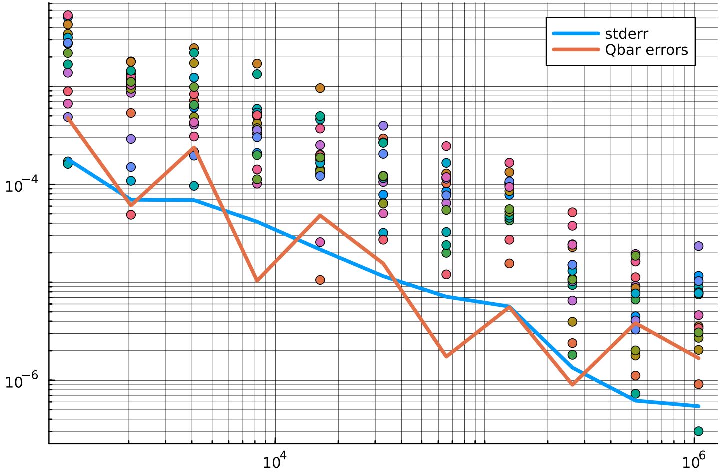
Random shifting gives us a practical error estimator:

```
shift_mod1(shift) = x -> (x + shift) .% 1

M = 10 # number of random shifts
shifts = rand(d, M)

# Gives an 11-by-10 matrix (M=10, number of powers of 2 = 11)
Qs = reduce(hcat,
            mean.(f o shift_mod1(shift), lr_seq(d, z, m1, m2))
            for shift in eachcol(shifts) )

Qbar = mean(Qs, dims=2)
std_err = std(Qs, dims=2) / M
```

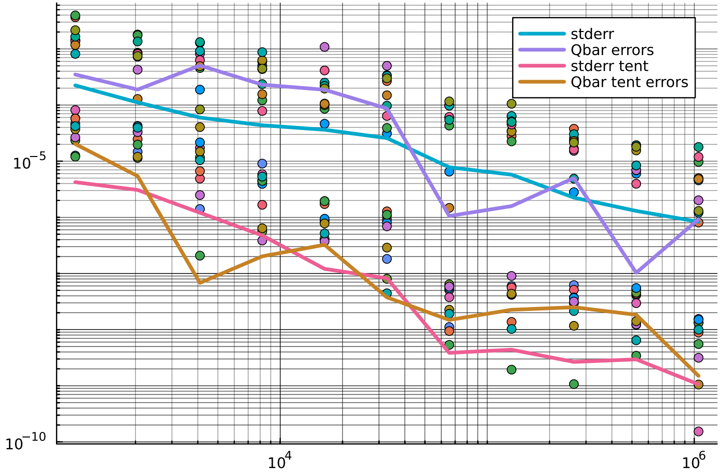
Standard error plot $d = 2$



Standard error plot $d = 5$



Standard error plot $d = 5$ and tent-transform



Lecture 2:
Weighted function spaces and tractability

4. Function spaces

Function spaces

- We express "smoothness" in terms of norms of derivatives.
- Such spaces are called Sobolev spaces.
- There is many ways of defining these norms / spaces.
- Even for "classical" Sobolev spaces.

- For high-dimensional spaces we will use "mixed" norms.
- For tractability we will add weights over the dimensions.

- We define spaces as being those functions for which the norm exists and is finite; and extra properties such as periodicity...

Classical Sobolev space $H^1(\Omega)$

Define the classical L^2 Sobolev space norm of order 1 as

$$
\begin{aligned}
\|u\|_{H^1(\Omega)} &:= \left(\|u\|_{L^2(\Omega)}^2 + \|\nabla u\|_{L^2(\Omega)}^2\right)^{1/2} \\
&= \left(\int_\Omega |u(\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{x} + \int_\Omega |\nabla u(\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{x}\right)^{1/2} \\
&= \left(\int_\Omega |u(\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{x} + \int_\Omega \sum_{j=1}^{d} |\partial_j u(\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{x}\right)^{1/2} \\
&= \left(\|u\|_{L^2(\Omega)}^2 + \sum_{\substack{\boldsymbol{\tau}\in\mathbb{N}_0^d \\ \|\boldsymbol{\tau}\|_1 = 1}} \|D^{\boldsymbol{\tau}} u\|_{L^2(\Omega)}^2\right)^{1/2} \\
&= \left(\sum_{\substack{\boldsymbol{\tau}\in\mathbb{N}_0^d \\ \|\boldsymbol{\tau}\|_1 \le 1}} \|D^{\boldsymbol{\tau}} u\|_{L^2(\Omega)}^2\right)^{1/2}.
\end{aligned}
$$

Classical L^2 Sobolev spaces $H^\alpha(\Omega)$

So

$$\|u\|_{H^1(\Omega)} := \Bigg(\sum_{\substack{\boldsymbol{\tau}\in\mathbb{N}_0^d \\ \|\boldsymbol{\tau}\|_1 \le 1}} \|D^{\boldsymbol{\tau}} u\|^2_{L^2(\Omega)} \Bigg)^{1/2}.$$

Define the corresponding space

$$H^1(\Omega) := \{u \in L_2(\Omega) : \|u\|_{H^1(\Omega)} < \infty\}.$$

For $\alpha \in \mathbb{N}_0$ define the Sobolev norm of order α by

$$\|u\|_{H^\alpha(\Omega)} := \Bigg(\sum_{\substack{\boldsymbol{\tau}\in\mathbb{N}_0^d \\ \|\boldsymbol{\tau}\|_1 \le \alpha}} \|D^{\boldsymbol{\tau}} u\|^2_{L^2(\Omega)} \Bigg)^{1/2},$$

and

$$H^\alpha(\Omega) := \left\{u \in L_2(\Omega) : \|u\|_{H^\alpha(\Omega)} < \infty\right\}.$$

NB: $H^0(\Omega) \equiv L^2(\Omega)$.

Classical $W^{\alpha,p}$ norms

Instead of taking L^2 norms, one can also take L^p norms.

For $\alpha \in \mathbb{N}_0$ and $1 \le p \le \infty$, define the norm

$$\|u\|_{W^{\alpha,p}(\Omega)} := \begin{cases} \left(\sum\limits_{\substack{\boldsymbol{\tau} \in \mathbb{N}_0^d \\ \|\boldsymbol{\tau}\|_1 \le \alpha}} \|D^{\boldsymbol{\tau}} u\|_{L^p(\Omega)}^p \right)^{1/p} & \text{for } 1 \le p < \infty, \\ \max\limits_{\substack{\boldsymbol{\tau} \in \mathbb{N}_0^d \\ \|\boldsymbol{\tau}\|_1 \le \alpha}} \|D^{\boldsymbol{\tau}} u\|_{L^\infty(\Omega)} & \text{for } p = \infty, \end{cases}$$

and the corresponding spaces

$$W^{\alpha,p}(\Omega) := \left\{ u \in L_p(\Omega) : \|u\|_{W^{\alpha,p}(\Omega)} < \infty \right\}.$$

Mixed Sobolev spaces

Classical norms have $\|\boldsymbol{\tau}\|_1 \leq \alpha$.

Mixed norms have $\|\boldsymbol{\tau}\|_\infty \leq \alpha$.

We will stick to the Hilbert space setting $p = 2$. Hence:

For $\alpha \in \mathbb{N}_0$ define the mixed Sobolev norm of order α by

$$\|u\|_{H^\alpha_{\mathrm{mix}}(\Omega)} := \Big(\sum_{\substack{\boldsymbol{\tau} \in \mathbb{N}_0^d \\ \|\boldsymbol{\tau}\|_\infty \leq \alpha}} \|D^{\boldsymbol{\tau}} u\|^2_{L^2(\Omega)} \Big)^{1/2},$$

and

$$H^\alpha_{\mathrm{mix}}(\Omega) := \big\{ u \in L_2(\Omega) : \|u\|_{H^\alpha_{\mathrm{mix}}(\Omega)} < \infty \big\}.$$

NB: $H^0_{\mathrm{mix}}(\Omega) \equiv H^0(\Omega) \equiv L^2(\Omega)$.
NBB: For $\Omega \subset \mathbb{R}$, i.e., $d = 1$: $H^\alpha_{\mathrm{mix}}(\Omega) \equiv H^\alpha(\Omega)$.

Fourier series and derivatives

Let us fix $\Omega = [0,1]^d$ and go back to periodic functions:

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{h} \in \mathbb{Z}^d} \hat{f}(\boldsymbol{h}) \exp(2\pi \mathrm{i}\, \boldsymbol{h} \cdot \boldsymbol{x}) = \sum_{\boldsymbol{h} \in \mathbb{Z}^d} \hat{f}(\boldsymbol{h}) \prod_{j=1}^{d} \mathrm{e}^{2\pi \mathrm{i}\, h_j x_j}.$$

We have, for $\boldsymbol{m} \in \mathbb{N}_0^d$, (and under sufficient smoothness conditions),

$$D^{\boldsymbol{m}} f(\boldsymbol{x}) =$$

Fourier series and derivatives

Let us fix $\Omega = [0,1]^d$ and go back to periodic functions:

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{h} \in \mathbb{Z}^d} \hat{f}(\boldsymbol{h}) \exp(2\pi\mathrm{i}\, \boldsymbol{h} \cdot \boldsymbol{x}) = \sum_{\boldsymbol{h} \in \mathbb{Z}^d} \hat{f}(\boldsymbol{h}) \prod_{j=1}^{d} \mathrm{e}^{2\pi\mathrm{i}\, h_j x_j}.$$

We have, for $\boldsymbol{m} \in \mathbb{N}_0^d$, (and under sufficient smoothness conditions),

$$D^{\boldsymbol{m}} f(\boldsymbol{x}) = \sum_{\boldsymbol{h} \in \mathbb{Z}^d} \hat{f}(\boldsymbol{h}) \Bigg[\prod_{\substack{j=1 \\ \text{s.t. } m_j \neq 0}}^{d} \frac{\partial^{m_j}}{\partial x_j^{m_j}} \mathrm{e}^{2\pi\mathrm{i}\, h_j x_j} \Bigg] \Bigg[\prod_{\substack{j=1 \\ \text{s.t. } m_j = 0}}^{d} \mathrm{e}^{2\pi\mathrm{i}\, h_j x_j} \Bigg]$$

Fourier series and derivatives

Let us fix $\Omega = [0,1]^d$ and go back to periodic functions:

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{h}\in\mathbb{Z}^d} \hat{f}(\boldsymbol{h}) \exp(2\pi\mathrm{i}\, \boldsymbol{h}\cdot\boldsymbol{x}) = \sum_{\boldsymbol{h}\in\mathbb{Z}^d} \hat{f}(\boldsymbol{h}) \prod_{j=1}^{d} \mathrm{e}^{2\pi\mathrm{i}\, h_j x_j}.$$

We have, for $\boldsymbol{m} \in \mathbb{N}_0^d$, (and under sufficient smoothness conditions),

$$D^{\boldsymbol{m}} f(\boldsymbol{x}) = \sum_{\boldsymbol{h}\in\mathbb{Z}^d} \hat{f}(\boldsymbol{h}) \Bigg[\prod_{\substack{j=1 \\ \text{s.t. } m_j\neq 0}}^{d} \frac{\partial^{m_j}}{\partial x_j^{m_j}} \mathrm{e}^{2\pi\mathrm{i}\, h_j x_j} \Bigg] \Bigg[\prod_{\substack{j=1 \\ \text{s.t. } m_j=0}}^{d} \mathrm{e}^{2\pi\mathrm{i}\, h_j x_j} \Bigg]$$

$$= \sum_{\substack{\boldsymbol{h}\in\mathbb{Z}^d \\ \text{s.t. } h_j\neq 0 \text{ when } m_j\neq 0}} \hat{f}(\boldsymbol{h}) \Bigg[\prod_{\substack{j=1 \\ \text{s.t. } m_j\neq 0}}^{d} (2\pi\mathrm{i}\, h_j)^{m_j} \Bigg] \mathrm{e}^{2\pi\mathrm{i}\, \boldsymbol{h}\cdot\boldsymbol{x}}.$$

Combine with integration...

Fix $\mathfrak{v} \subseteq \{1, \ldots, d\}$, then

$$I_{\mathfrak{v}}(f) := \int_{[0,1]^{|\mathfrak{v}|}} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}_{\mathfrak{v}} = \sum_{\boldsymbol{h} \in \mathbb{Z}^d} \hat{f}(\boldsymbol{h}) \Big[\prod_{j \in \mathfrak{v}} \int_0^1 \mathrm{e}^{2\pi\mathrm{i}\, h_j x_j} \, \mathrm{d}x_j \Big] \Big[\prod_{j \notin \mathfrak{v}} \mathrm{e}^{2\pi\mathrm{i}\, h_j x_j} \Big]$$

$$= \sum_{\substack{\boldsymbol{h} \in \mathbb{Z}^d \\ \text{s.t. } h_j = 0 \text{ when } j \in \mathfrak{v}}} \hat{f}(\boldsymbol{h}) \exp(2\pi\mathrm{i}\, \boldsymbol{h} \cdot \boldsymbol{x}).$$

Combine with integration...

Fix $\mathfrak{v} \subseteq \{1, \ldots, d\}$, then

$$I_{\mathfrak{v}}(f) := \int_{[0,1]^{|\mathfrak{v}|}} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}_{\mathfrak{v}} = \sum_{\boldsymbol{h} \in \mathbb{Z}^d} \hat{f}(\boldsymbol{h}) \Big[\prod_{j \in \mathfrak{v}} \int_0^1 \mathrm{e}^{2\pi \mathrm{i}\, h_j x_j} \, \mathrm{d}x_j \Big] \Big[\prod_{j \notin \mathfrak{v}} \mathrm{e}^{2\pi \mathrm{i}\, h_j x_j} \Big]$$

$$= \sum_{\substack{\boldsymbol{h} \in \mathbb{Z}^d \\ \text{s.t. } h_j = 0 \text{ when } j \in \mathfrak{v}}} \hat{f}(\boldsymbol{h}) \exp(2\pi \mathrm{i}\, \boldsymbol{h} \cdot \boldsymbol{x}).$$

We had

$$D^{\boldsymbol{m}} f(\boldsymbol{x}) = \sum_{\substack{\boldsymbol{h} \in \mathbb{Z}^d \\ \text{s.t. } h_j \neq 0 \text{ when } j \in \mathrm{supp}(\boldsymbol{m})}} \hat{f}(\boldsymbol{h}) \Big[\prod_{j \in \mathrm{supp}(\boldsymbol{m})} (2\pi \mathrm{i}\, h_j)^{m_j} \Big] \mathrm{e}^{2\pi \mathrm{i}\, \boldsymbol{h} \cdot \boldsymbol{x}}.$$

Combine with integration...

Fix $\mathfrak{v} \subseteq \{1, \ldots, d\}$, then

$$I_{\mathfrak{v}}(f) := \int_{[0,1]^{|\mathfrak{v}|}} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}_{\mathfrak{v}} = \sum_{\boldsymbol{h} \in \mathbb{Z}^d} \hat{f}(\boldsymbol{h}) \Big[\prod_{j \in \mathfrak{v}} \int_0^1 \mathrm{e}^{2\pi\mathrm{i}\, h_j x_j} \, \mathrm{d}x_j \Big] \Big[\prod_{j \notin \mathfrak{v}} \mathrm{e}^{2\pi\mathrm{i}\, h_j x_j} \Big]$$

$$= \sum_{\substack{\boldsymbol{h} \in \mathbb{Z}^d \\ \text{s.t. } h_j = 0 \text{ when } j \in \mathfrak{v}}} \hat{f}(\boldsymbol{h}) \exp(2\pi\mathrm{i}\, \boldsymbol{h} \cdot \boldsymbol{x}).$$

We had

$$D^{\boldsymbol{m}} f(\boldsymbol{x}) = \sum_{\substack{\boldsymbol{h} \in \mathbb{Z}^d \\ \text{s.t. } h_j \neq 0 \text{ when } j \in \mathrm{supp}(\boldsymbol{m})}} \hat{f}(\boldsymbol{h}) \Big[\prod_{j \in \mathrm{supp}(\boldsymbol{m})} (2\pi\mathrm{i}\, h_j)^{m_j} \Big] \mathrm{e}^{2\pi\mathrm{i}\, \boldsymbol{h} \cdot \boldsymbol{x}}.$$

Define $\mathrm{supp}^*(\boldsymbol{m}) := \{1, \ldots, d\} \setminus \mathrm{supp}(\boldsymbol{m})$, then

$$I_{\mathrm{supp}^*(\boldsymbol{m})} D^{\boldsymbol{m}} f(\boldsymbol{x}) = \sum_{\substack{\boldsymbol{h} \in \mathbb{Z}^d \\ \text{s.t. } \mathrm{supp}(\boldsymbol{h}) = \mathrm{supp}(\boldsymbol{m})}} \hat{f}(\boldsymbol{h}) \Big[\prod_{j \in \mathrm{supp}(\boldsymbol{h})} (2\pi\mathrm{i}\, h_j)^{m_j} \Big] \mathrm{e}^{2\pi\mathrm{i}\, \boldsymbol{h} \cdot \boldsymbol{x}}.$$

Finally, the norm...

Note that

$$\sum_{\boldsymbol{h}\in\mathbb{Z}^d} A(\boldsymbol{h}) = \sum_{\mathfrak{u}\subseteq\{1,\dots,d\}} \sum_{\substack{\boldsymbol{h}\in\mathbb{Z}^d \\ \text{s.t. } \operatorname{supp}(\boldsymbol{h})=\mathfrak{u}}} A(\boldsymbol{h}).$$

Using Parseval, it follows that

$$\begin{aligned}
\sum_{\boldsymbol{m}\in\{0,\alpha\}^d} \|I_{\operatorname{supp}^*(\boldsymbol{m})} D^{\boldsymbol{m}} f\|_{L_2}^2 &= \sum_{\mathfrak{u}\subseteq\{1,\dots,d\}} \|I_{\{1,\dots,d\}\setminus\mathfrak{u}} D^{\boldsymbol{\alpha}_{\mathfrak{u}}} f\|_{L_2}^2 \\
&= \sum_{\mathfrak{u}\subseteq\{1,\dots,d\}} \sum_{\substack{\boldsymbol{h}\in\mathbb{Z}^d \\ \text{s.t. } \operatorname{supp}(\boldsymbol{h})=\mathfrak{u}}} |\hat{f}(\boldsymbol{h})|^2 \prod_{j\in\mathfrak{u}} |2\pi h_j|^{2\alpha} \\
&= \sum_{\boldsymbol{h}\in\mathbb{Z}^d} |\hat{f}(\boldsymbol{h})|^2 \prod_{j\in\operatorname{supp}(\boldsymbol{h})} |2\pi h_j|^{2\alpha}.
\end{aligned}$$

Yesterday with $d = 1$ we had $\|f^{(\alpha)}\|_{L_2}$ *semi*-norm...

5. Error bounds for lattice rules

Cubature error for lattice rule

Remember: lattice rule = average over lattice points $(k\boldsymbol{z} \bmod n)/n$.
For periodic function, + smoothness conditions:

$$\frac{1}{n}\sum_{k\in\mathbb{Z}_n} f(\boldsymbol{z}k/n) - I(f) = \sum_{\mathbf{0}\neq\boldsymbol{h}\in\mathbb{Z}^d} \hat{f}(\boldsymbol{h})\,\frac{1}{n}\sum_{k\in\mathbb{Z}_n} \exp(2\pi\mathrm{i}\,(\boldsymbol{h}\cdot\boldsymbol{z})\,k/n)$$
$$= \sum_{\substack{\mathbf{0}\neq\boldsymbol{h}\in\mathbb{Z}^d \\ \boldsymbol{h}\cdot\boldsymbol{z}\equiv 0 \pmod n}} \hat{f}(\boldsymbol{h}).$$

Error consists of the Fourier coefficients for which $\boldsymbol{h}\cdot\boldsymbol{z}$ is a multiple of n, except $\boldsymbol{h}=\mathbf{0}$.

Error bound

Hence, for $\alpha > 1/2$,

$$\left|\frac{1}{n}\sum_{k\in\mathbb{Z}_n} f(\boldsymbol{z}k/n) - I(f)\right| = \left|\sum_{\substack{\boldsymbol{0}\neq\boldsymbol{h}\in\mathbb{Z}^d\\ \boldsymbol{h}\cdot\boldsymbol{z}\equiv 0 \pmod n}} \hat{f}(\boldsymbol{h})\right|$$

$$= \left|\sum_{\substack{\boldsymbol{0}\neq\boldsymbol{h}\in\mathbb{Z}^d\\ \boldsymbol{h}\cdot\boldsymbol{z}\equiv 0 \pmod n}} \hat{f}(\boldsymbol{h}) \frac{\prod_{j\in\mathrm{supp}(\boldsymbol{h})} |2\pi h_j|^\alpha}{\prod_{j\in\mathrm{supp}(\boldsymbol{h})} |2\pi h_j|^\alpha}\right|$$

$$\leq \Bigg(\sum_{\boldsymbol{h}\in\mathbb{Z}^d} |\hat{f}(\boldsymbol{h})|^2 \prod_{j\in\mathrm{supp}(\boldsymbol{h})} |2\pi h_j|^{2\alpha}\Bigg)^{1/2} \Bigg(\sum_{\substack{\boldsymbol{0}\neq\boldsymbol{h}\in\mathbb{Z}^d\\ \boldsymbol{h}\cdot\boldsymbol{z}\equiv 0 \pmod n}} \prod_{j\in\mathrm{supp}(\boldsymbol{h})} |2\pi h_j|^{-2\alpha}\Bigg)^{1/2}$$

$$= \|f\|_\alpha^2 \, e(\alpha, \boldsymbol{z}, n).$$

We implicitly defined a norm, recognizable from previous slides.
Now we need to show that $e(\alpha, \boldsymbol{z}, n)$ is of order $n^{-\alpha}$.

Function space, including weights

Korobov space of dominating mixed smoothness $\alpha > 1/2$:

$$\mathcal{H}_\alpha := \left\{ f \in L_2([0,1]^d) : \|f\|_\alpha^2 := \sum_{\boldsymbol{h} \in \mathbb{Z}^d} r_\alpha^2(\boldsymbol{h}) \, |\hat{f}(\boldsymbol{h})|^2 < \infty \right\},$$

with

$$r_\alpha(\boldsymbol{h}) := \gamma_{\mathrm{supp}(\boldsymbol{h})}^{-1} \prod_{j \in \mathrm{supp}(\boldsymbol{h})} |h_j|^\alpha.$$

Weighted spaces: Sloan & Woźniakowski (2001),
Novak & Woźniakowski (2008, 2010, 2012), ...

We need the weights to get error bounds independent of d.

6. Worst-case error analysis

How to measure deterministic algorithms? (Intro to IBC)

- Worst-case error for approximating $I(f)$ by $Q_n(f)$ for $f \in \mathcal{F}_d$:

$$e(Q_n, \mathcal{H}_{d,\alpha,\gamma}) := \sup_{\substack{f \in \mathcal{H}_{d,\alpha,\gamma} \\ \|f\|_{d,\alpha,\gamma} \leq 1}} |I(f) - Q_n(f)| \;\leq\; \text{upper bound for } Q_n.$$

- Best possible error using n function values (benchmark):

$$e(n, \mathcal{H}_{d,\alpha,\gamma}) := \inf_{Q_n : \{(w_k, \boldsymbol{x}_k)\}_{k=1}^n} e(Q_n; \mathcal{H}_{d,\alpha,\gamma}) \;\geq\; \text{lower bound for any}$$

$$= \textit{error of best algorithm using n function evaluations.}$$

- Information complexity: the minimal number of function values needed to reach error at most ϵ:

$$n(\epsilon, \mathcal{H}_{d,\alpha,\gamma}) := \min\{n : \exists Q_n \text{ for which } e(Q_n, \mathcal{H}_{d,\alpha,\gamma}) \leq \epsilon\}$$

$$= \textit{number of function evaluations of best algorithm.}$$

See a multitude of references, e.g., Novak (2016) or the Novak–Woźniakowski trilogy (2008,2010,2012), . . .

The curse of dimensionality & types of tractability

Tractability started by Woźniakowski (1994) and since then vastly expanded...

- The curse of dimensionality is defined as needing an exponential number of function values in d to reach an error $\epsilon \leq \epsilon_0$:

$$n(\epsilon, \mathcal{H}_{d,\alpha,\gamma}) \geq c\,(1+\gamma)^d, \qquad \text{for some } c, \gamma, \epsilon_0 > 0.$$

- A problem is called (weakly) tractable if

$$\lim_{\epsilon^{-1}+d\to\infty} \frac{\ln n(\epsilon, d)}{\epsilon^{-1} + d} = 0,$$

and intractable otherwise.

- Different types, e.g., polynomial tractability

$$n(\epsilon, \mathcal{H}_{d,\alpha,\gamma}) \leq c\,\epsilon^{-p}\,d^q, \qquad \text{for some } c, p, q \geq 0.$$

See a multitude of references, in particular the Novak–Woźniakowski trilogy (2008,2010,2012), ...

The curse might always be there...

Define $\mathcal{F}_d$ with $f \in \mathcal{F}_d$ when

$$\|f\|_{\mathcal{F}_d} := \max_{\boldsymbol{x},\boldsymbol{y}\in[0,1]^d} \frac{|f(\boldsymbol{x}) - f(\boldsymbol{y})|}{\|\boldsymbol{x} - \boldsymbol{y}\|_\infty} \quad < \quad \infty,$$

then (Maung Zho Newn and Sharygin, 1971)

$$e(n, \mathcal{F}_d) = \frac{d}{2d+2}\, n^{-1/d}.$$

This is for any (linear) algorithm!
See also Novak (2016).
The aim is to not just avoid the "curse by construction" (product rule $n = m^d$), but also

- rate independent of d $\Rightarrow$ "mixed dominating smoothness".
- constant $C_{d,\alpha,\gamma}$ independent of d $\Rightarrow$ "weighted spaces".

Tools / assumptions

- **Mixed dominating smoothness spaces:**
  Classical Sobolev norm with $\|\boldsymbol{\tau}\|_1 \le \alpha$ gives $O(n^{-\alpha/d})$; mixed norm with $\|\boldsymbol{\tau}\|_\infty \le \alpha$ gives $O(n^{-\alpha})$. I.e.,

$$\sum_{\substack{\boldsymbol{\tau}\in\{0,\ldots,\alpha\}^d \\ \|\boldsymbol{\tau}\|_\infty\le\alpha}} \|D^{\boldsymbol{\tau}} f\|_{L_2}^2 \quad \text{versus} \quad \sum_{\substack{\boldsymbol{\tau}\in\{0,\ldots,\alpha\}^d \\ \|\boldsymbol{\tau}\|_1\le\alpha}} \|D^{\boldsymbol{\tau}} f\|_{L_2}^2.$$

- **Dimension-independent error bounds:**
  Switch to weighted spaces: not all combinations of variables are as important. Denote the importance of the variables in $\mathfrak{u} \subseteq \{1, \ldots, d\}$ by $\gamma_{\mathfrak{u}}$. I.e.,

$$\sum_{\substack{\boldsymbol{\tau}\in\{0,\ldots,\alpha\}^d \\ \|\boldsymbol{\tau}\|_\infty\le\alpha}} \gamma_{\mathrm{supp}(\boldsymbol{\tau})}^{-1} \|D^{\boldsymbol{\tau}} f\|_{L_2}^2.$$

Mixed spaces: Novak, Sickel, Temlyakov, Kühn, Ullrich, Ullrich, Potts, ...
Weights: Hickernell (1998), Sloan & Woźniakowski (1998), Novak–Woźniakowski, Dick, Kuo, Sloan (2013), ...

Again our favourite function space

Korobov space of dominating mixed smoothness $\alpha > 1/2$:

$$\mathcal{H}_{d,\alpha,\gamma} := \left\{ f \in L_2([0,1]^d) : \|f\|_{d,\alpha,\gamma}^2 < \infty \right\},$$

with

$$\|f\|_{d,\alpha,\gamma}^2 := \sum_{\boldsymbol{h} \in \mathbb{Z}^s} r_{d,\alpha,\gamma}^2(\boldsymbol{h}) \, |\hat{f}(\boldsymbol{h})|^2$$

and

$$r_{d,\alpha,\gamma}^2(\boldsymbol{h}) := \gamma_{\mathrm{supp}(\boldsymbol{h})}^{-1} \prod_{j \in \mathrm{supp}(\boldsymbol{h})} |h_j|^{2\alpha}.$$

(Sometimes the 2π is present, sometimes it is not.)
(Sometimes the 2α is taken as α, different convention.)
(Sometimes the weights are squared.)

For integer smoothness

When $\alpha \in \mathbb{N}$ then this norm can be written as the norm of the unanchored periodic Sobolev space of dominating mixed smoothness α:

$$\|f\|_{d,\alpha,\boldsymbol{\gamma}}^2 := \sum_{\boldsymbol{h}\in\mathbb{Z}^d} r_{d,\alpha,\boldsymbol{\gamma}}^2(\boldsymbol{h})\,|\hat{f}(\boldsymbol{h})|^2 = \sum_{\boldsymbol{h}\in\mathbb{Z}^d} \gamma_{\mathrm{supp}(\boldsymbol{h})}^{-1}\,|\hat{f}(\boldsymbol{h})|^2 \prod_{j\in\mathrm{supp}(\boldsymbol{h})} |h_j|^{2\alpha}$$

$$= \sum_{\substack{\boldsymbol{\nu}\in\{0,\alpha\}^d \\ \mathfrak{u}:=\mathrm{supp}(\boldsymbol{\nu})}} \frac{\gamma_{\mathfrak{u}}^{-1}}{\prod_{j\in\mathfrak{u}}(2\pi)^{2\nu_j}} \int_{[0,1]^{|\mathfrak{u}|}} \underbrace{\left|\int_{[0,1]^{d-|\mathfrak{u}|}} f^{(\boldsymbol{\nu})}(\boldsymbol{y}_{-\mathfrak{u}},\boldsymbol{y}_{\mathfrak{u}})\,\mathrm{d}\boldsymbol{y}_{-\mathfrak{u}}\right|}_{\text{“unanchored”}}{}^2\,\mathrm{d}\boldsymbol{y}_{\mathfrak{u}}$$

$$= \sum_{\substack{\boldsymbol{\nu}\in\{0,\alpha\}^d \\ \mathfrak{u}:=\mathrm{supp}(\boldsymbol{\nu})}} \gamma_{\mathfrak{u}}^{-1}\,\|I_{-\mathfrak{u}}\,f^{(\boldsymbol{\nu})}\|_{L_2}^2.$$

Usual error bounds

Example theorem.
For $f \in \mathcal{H}_{d,\alpha,\gamma}$ with $\alpha > 1/2$ and $n \in \mathbb{N}$ we can construct a generating vector $\boldsymbol{z} \in \mathbb{Z}_n^d$ such that

$$|I(f) - Q_{n,\boldsymbol{z}}(f)| \leq \frac{C_{d,\alpha,\gamma,\lambda}}{n^\lambda} \|f\|_{d,\alpha,\gamma} \qquad \text{for all } \lambda \in [1/2, \alpha)$$

with

$$C_{d,\alpha,\gamma,\lambda} = \dots$$

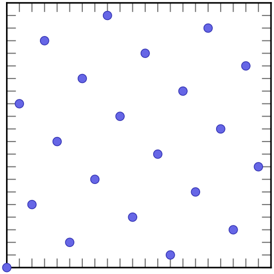
With the right summability conditions on the weights this becomes a dimension-independent convergence bound for some $C'_{\alpha,\gamma,\lambda}$ with $C_{d,\alpha,\gamma,\lambda} < C'_{\alpha,\gamma,\lambda} < \infty$.

See a lot of CBC and fast CBC papers: Kuo, Sloan, Dick, Nuyens, Kritzer, Ebert, Wilkes, Schwab, ...

7. Reproducing kernel Hilbert spaces

Example of a good lattice rule

Eg: $n = 21$ and $\boldsymbol{z} = (1, 13)$: Fibonacci rule: $n = F_k$, $\boldsymbol{z} = (1, F_{k-1})$.



Only $d = 2$, $d \geq 2$: Constructive methods for deterministic error:
Fast component-by-component (Nuyens & Cools 2006, ...)
$\rightarrow$ Fixed vector $\boldsymbol{z}$ for a given n.
(Or sequence of $n = p^m$, Cools, Kuo & Nuyens 2006).

Spaces based on series representations & Koksma–Hlawka

Assume L_2-ONB $\{\phi_h\}_h$, $\phi_0 = 1$, $Q_n(1) = 1$, and abs. summ.

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{h}} \hat{f}(\boldsymbol{h})\, \phi_{\boldsymbol{h}}(\boldsymbol{x}), \qquad \text{with} \qquad \hat{f}(\boldsymbol{h}) := \int_{[0,1]^d} f(\boldsymbol{x})\, \overline{\phi_{\boldsymbol{h}}(\boldsymbol{x})}\, \mathrm{d}\boldsymbol{x},$$

Spaces based on series representations & Koksma–Hlawka

Assume L_2-ONB $\{\phi_h\}_h$, $\phi_0 = 1$, $Q_n(1) = 1$, and abs. summ.

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{h}} \hat{f}(\boldsymbol{h})\, \phi_{\boldsymbol{h}}(\boldsymbol{x}), \qquad \text{with} \qquad \hat{f}(\boldsymbol{h}) := \int_{[0,1]^d} f(\boldsymbol{x})\, \overline{\phi_{\boldsymbol{h}}(\boldsymbol{x})}\, \mathrm{d}\boldsymbol{x},$$

then, for $r_{\alpha,\gamma}(\boldsymbol{h}) > 0$ an "increasing" function,

$$|I(f) - Q_n(f)| = \left| \sum_{\boldsymbol{h} \neq \boldsymbol{0}} \hat{f}(\boldsymbol{h})\, Q_n(\phi_{\boldsymbol{h}})\, r_{\alpha,\gamma}(\boldsymbol{h})\, r_{\alpha,\gamma}^{-1}(\boldsymbol{h}) \right|$$

$$\leq \underbrace{\left(\sum_{\boldsymbol{h}} \left| \hat{f}(\boldsymbol{h}) \right|^p r_{\alpha,\gamma}^p(\boldsymbol{h}) \right)^{1/p}}_{\text{norm}} \times \underbrace{\left(\sum_{\boldsymbol{h} \neq \boldsymbol{0}} |Q_n(\phi_{\boldsymbol{h}})|^q\, r_{\alpha,\gamma}^{-q}(\boldsymbol{h}) \right)^{1/q}}_{\text{worst-case error}^*} .$$

(See next slide.)

Worst-case error (continued...)

$$|I(f) - Q_n(f)| = \left| \sum_{\boldsymbol{h} \neq \boldsymbol{0}} \hat{f}(\boldsymbol{h})\, Q_n(\phi_{\boldsymbol{h}})\, r_{\alpha,\gamma}(\boldsymbol{h})\, r_{\alpha,\gamma}^{-1}(\boldsymbol{h}) \right|$$

$$\leq \underbrace{\left(\sum_{\boldsymbol{h}} \left| \hat{f}(\boldsymbol{h}) \right|^p r_{\alpha,\gamma}^p(\boldsymbol{h}) \right)^{1/p}}_{\text{norm}} \times \underbrace{\left(\sum_{\boldsymbol{h} \neq \boldsymbol{0}} |Q_n(\phi_{\boldsymbol{h}})|^q\, r_{\alpha,\gamma}^{-q}(\boldsymbol{h}) \right)^{1/q}}_{\text{worst-case error}^*} .$$

For $1 < p \leq \infty$ and compatible choices of $\phi_{\boldsymbol{h}}$, Q_n and $r_{\alpha,\gamma}$ we can find a "worst-case" representer $\xi(\boldsymbol{x})$ for which

$$|Q_n(\xi) - I(\xi)|^{1/q} = e(Q_n, \mathcal{F}_d), \qquad (*)$$

independent of the particular Q_n, e.g., Fourier series and lattice rules, Walsh series and digital nets, see Nuyens (2014) and Hickernell (1998a,b).

Reproducing kernel Hilbert spaces, $p = q = 2$

Given a one-dimensional reproducing kernel $K(x, y) = \overline{K(y, x)}$.
Suppose $\mathcal{H}(K)$ is separable: $\mathcal{H}(K) = \operatorname{span}\{\phi_h\}_h$ and $\phi_0 = 1$.
Determine the eigenvalues and eigenfunctions, and assume $\lambda_0 = 1$,

$$\int_{[0,1]} \phi(x)\, \overline{K(x, y)}\, \mathrm{d}x = \lambda\, \phi(y).$$

Then

$$K(x, y) = \sum_h \frac{\phi_h(x)}{\sqrt{\lambda_h}}\, \frac{\overline{\phi_h(y)}}{\sqrt{\lambda_h}} = \sum_h \frac{\phi_h(x)}{\|\phi_h\|_{L_2}}\, \frac{\overline{\phi_h(y)}}{\|\phi_h\|_{L_2}},$$

the ϕ_h are L_2-orthogonal, with $\|\phi_h\|_{L_2} = \sqrt{\lambda_h}$ and $\|\phi_h\|_{\mathcal{H}} = 1$, with

$$\langle f, g\rangle_{\mathcal{H}} = \sum_h \lambda_h\, \hat{f}(h)\, \overline{\hat{g}(h)}, \qquad \|f\|_{\mathcal{H}}^2 = \sum_h \lambda_h \left|\hat{f}(h)\right|^2.$$

Multivariate weighted reproducing kernel Hilbert space

Use the one-dimensional space as building block for d dimensions by taking weighted tensor products (tensor product basis):

$$K(\boldsymbol{x},\boldsymbol{y}) = \sum_{\mathfrak{u}\subseteq\{1,\ldots,d\}} \gamma_{\mathfrak{u}} \prod_{j\in\mathfrak{u}} K(x_j,y_j) = \sum_{\boldsymbol{h}} \gamma_{\mathrm{supp}(\boldsymbol{h})} \prod_{j=1}^{d} \frac{\phi_{h_j}(x_j)}{\sqrt{\lambda_{h_j}}} \frac{\overline{\phi_{h_j}(y_j)}}{\sqrt{\lambda_{h_j}}}$$

$$= \sum_{\boldsymbol{h}} r_{\alpha,\boldsymbol{\gamma}}^{-2}(\boldsymbol{h})\, \phi_{\boldsymbol{h}}(\boldsymbol{x})\, \overline{\phi_{\boldsymbol{h}}(\boldsymbol{y})},$$

with

$$r_{\alpha,\boldsymbol{\gamma}}^{-2}(\boldsymbol{h}) = \gamma_{\mathrm{supp}(\boldsymbol{h})} \prod_{j=1}^{d} \lambda_{h_j}^{-1}.$$

With $\gamma_\emptyset = 1$ and $Q_n(1) = 1$,

$$e^2(Q_n;\mathcal{H}) = -1 + \sum_{k,\ell=1}^{n} w_k\, w_\ell\, K(\boldsymbol{x}_k, \boldsymbol{y}_\ell).$$

For a shift-invariant space and lattice rule

For a shift-invariant space we have

$$K(\boldsymbol{x}, \boldsymbol{y}) = K(\boldsymbol{x} - \boldsymbol{y}, \boldsymbol{0})$$

and for a lattice rule we have

$$\boldsymbol{x}_k - \boldsymbol{x}_{k'} = \boldsymbol{x}_{k-k' \bmod n},$$

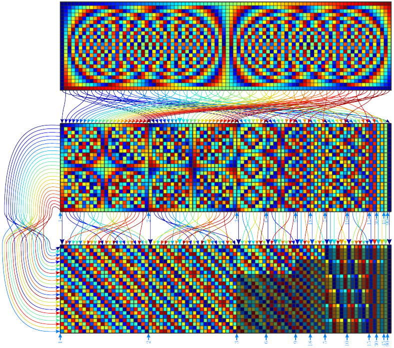
all on the torus $[0,1)^d$.
Hence:

$$\begin{aligned} e^2(Q_{n,\boldsymbol{z}}; \mathcal{H}) &= -1 + \sum_{k,\ell=1}^{n} w_k\, w_\ell\, K(\boldsymbol{x}_k, \boldsymbol{y}_\ell) \\ &= -1 + \sum_{\ell=1}^{n} \frac{1}{n} \sum_{k=1}^{n} \frac{1}{n} K(\boldsymbol{x}_{k-\ell \bmod n}, \boldsymbol{0}) \\ &= -1 + \frac{1}{n} \sum_{k=1}^{n} K(\boldsymbol{x}_k, \boldsymbol{0}). \end{aligned}$$

8. Fast component-by-component construction of good lattice rules

Construction of lattice rules and polynomial lattice rules

Point sets constructed for weighted spaces using fast component-by-component constructions using number theoretic transforms.



See https://www.cs.kuleuven.be/~dirkn/qmc4pde/ and https://www.cs.kuleuven.be/~dirkn/fast-cbc/.

See, e.g., Nuyens & Cools (2006a,2006b), Cools, Kuo, & Nuyens (2006), Dick, Kuo, Le Gia, Nuyens & Schwab (2014), Nuyens (2014), Kuo & Nuyens (2016), . . .
Variations and speedups by: Gantner, Kritzer, Laimer, Leobacher, Pillichshammer, Schwab, . . . New methods: Ebert, Kritzer, Nuyens, Osisiogu (2021), Kuo, Nuyens, Wilkes (2023), Nuyens, Wilkes (2023), . . .

Point generators

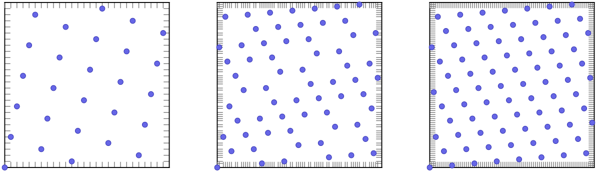
- Matlab/Octave: procedural generators like Matlab's `rand`:
  - `latticeseq_b2.m`: radical inverse lattice sequence generator,
  - `digitalseq_b2g.m`: gray coded radical inverse digital sequence generator (incl. higher-order, max 53 bit).
- Python: iterator classes, which can be used as standalone point generators from the command line (`__main__`):
  - `latticeseq_b2.py`: iterator based (`__iter__`), `set_state` for parallel computing,
  - `digitalseq_b2g.py`: ditto, arbitrary precision using `mpmath` if needed.
- C++: header file based implementation with driver program for the command line:
  - `latticeseq_b2.(h|cpp)`: complies to `ForwardIterator` concept, `set_state` for parallel computing,
  - `digitalseq_b2g.(h|cpp)`: ditto, max 64 bit.

Welcome to *The Magic Point Shop!*

Different flavours of quasi-Monte Carlo points to choose:

- Lattice rules.
- Lattice sequences.
- Polynomial lattice rules.
- Interlaced Sobol' sequences (higher-order).
- Interlaced polynomial lattice rules (higher-order).

And code (C++, Python and Matlab) to use them...



Subsidiaries: **QMC4PDE**: construct points for parametrised PDEs.

Lecture 3:
Advanced topics

9. Applications in uncertainty quantification (UQ)

High-dimensional integrals for $G(u(\boldsymbol{x}, \boldsymbol{y}))$

Task: Approximate an s-dimensional integral / expectation

$$\mathbb{E}[G(u)] = I(G(u)) := \int_{\mathbb{R}^s} G(u(\boldsymbol{y}))\, p(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}$$
$$= \int_{[0,1]^s} G(u(P^{-1}(\boldsymbol{y})))\, \mathrm{d}\boldsymbol{y}.$$

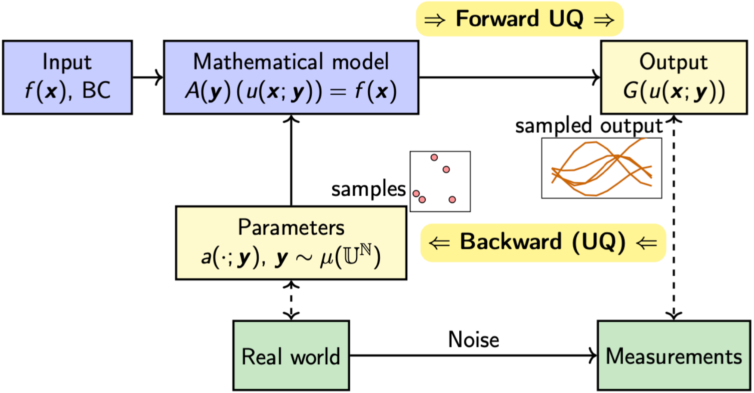
Method: An n-point cubature/quadrature method

$$Q_n(G(u)) = Q_n(G(u); \{(w_k, \boldsymbol{y}_k)\}_{k=1}^n) := \sum_{k=1}^{n} w_k\, G(u(\boldsymbol{y}_k)).$$

(Using functional analysis and number theoretic uniform point sets.
To tackle integration, approximation and "other" high dimensional problems.)

Applications: random fields, parametrised PDEs, financial engineering, Bayesian integrals, uncertainty quantification,. . .

Uncertainty quantification



QMC for high-dimensional integrals,
use s for number of (truncated) "stochastic" dimensions:

- Forward UQ: expected value of a quantity of interest.
- Backward UQ: estimate parameter values by Bayesian integrals.

(QMC can also be used for function approximation.)

Truncation, discretization, cubature: three errors

1. Truncate after s terms

$$a^s(\boldsymbol{x};\boldsymbol{y}) = a_0(\boldsymbol{x}) + \sum_{j=1}^{s} y_j\,\varphi_j(\boldsymbol{x}),$$

   then the solution u^s is the solution of the truncated problem.
2. Discretize the PDE: use FEM and discretize with elements of diameter h. The discretized solution we denote by u_h^s.
3. Cubature approximation of integral:

$$\frac{1}{n}\sum_{k=1}^{n} G(u_h^s(\cdot;\boldsymbol{t}^{(k)})).$$

$\Rightarrow$ Total error is the sum of three errors.

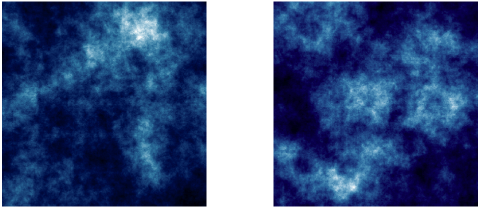
Example: random fields / parametrised PDEs ($s = \infty$)

Parametric representation (e.g., Karhunen–Loève expansion)

$$a(\boldsymbol{x}, \boldsymbol{y}) = a_0(\boldsymbol{x}) + \sum_{j \geq 1} y_j \, \psi_j(\boldsymbol{x}), \quad \boldsymbol{x} \in D, \quad \boldsymbol{y} \in [-\tfrac{1}{2}, \tfrac{1}{2}]^{\mathbb{N}},$$

by sample variables y_j. Use in porous flow using Darcy's law:

$$\begin{aligned} q(\boldsymbol{x}, \boldsymbol{y}) + a(\boldsymbol{x}, \boldsymbol{y}) \nabla p(\boldsymbol{x}, \boldsymbol{y}) &= f(\boldsymbol{x}), \\ \nabla \cdot q(\boldsymbol{x}, \boldsymbol{y}) &= 0. \end{aligned}$$



See, e.g., Barth, Charrier, Cliffe, Dick, Gantner, Giles, Graham, Haji-Ali, Harbrecht, Kuo, Le Gia, Nuyens, Nichols, Nobile, Peters, Robbe, Scheichl, Schwab, Siebenmorgen, Sloan, Teckentrup, Tempone, Ullmann, Vandewalle, Zollinger, von Schwerin, ...

Example: option pricing ($s =$ hundreds, thousands, ∞)

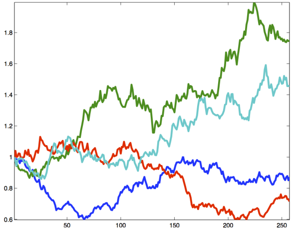
Simulation of SDE

$$dX(t) = a(X(t))\,dt + b(X(t))\,dW(t), \quad X(0) = X_0, \quad 0 \le t \le T,$$

using Euler–Maruyama, $\hat{X}_0 = X_0$,

$$\hat{X}_{i+1} = \hat{X}_i + a(\hat{X}_i)\,h + b(\hat{X}_i)\,\sqrt{h}\,Z_i, \quad i = 1, \ldots, n-1, \quad h = T/n,$$

with Z_i sampled from standard normal distribution.



See, e.g., Achtsis, Baldeaux, Boyle, Cools, Gerstner, Giles, Glasserman, Griebel, Holtz, Imai, Irrgeher, Joshi, Kucherenko, Kuo, L'Écuyer, Larcher, Lemieux, Leobacher, Lin, Nuyens, Niu, Ökten, Pages, Platen, Sloan, Staum, Szölgyenyi, Tan, Tezuka, Tichy, Traub, Tuffin, Wang, Waterhouse, ...

Example: option pricing ($s =$ hundreds, thousands, ∞)

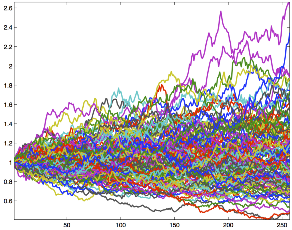
Simulation of SDE

$$\mathrm{d}X(t) = a(X(t))\,\mathrm{d}t + b(X(t))\,\mathrm{d}W(t), \quad X(0) = X_0, \quad 0 \le t \le T,$$

using Euler–Maruyama, $\hat{X}_0 = X_0$,

$$\hat{X}_{i+1} = \hat{X}_i + a(\hat{X}_i)\,h + b(\hat{X}_i)\,\sqrt{h}\,Z_i, \quad i = 1, \ldots, n-1, \quad h = T/n,$$

with Z_i sampled from standard normal distribution.



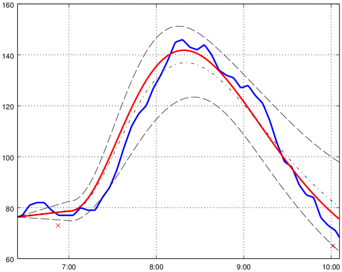
See, e.g., Achtsis, Baldeaux, Boyle, Cools, Gerstner, Giles, Glasserman, Griebel, Holtz, Imai, Irrgeher, Joshi, Kucherenko, Kuo, L'Écuyer, Larcher, Lemieux, Leobacher, Lin, Nuyens, Niu, Ökten, Pages, Platen, Sloan, Staum, Szölgyenyi, Tan, Tezuka, Tichy, Traub, Tuffin, Wang, Waterhouse, ...

Example: Bayesian integrals (s = hundreds, thousands, ∞)

Simulation of insulin-glucose model

$$\mathrm{d}G(t)/\mathrm{d}t = -\lambda\,(G(t) - G_b) - \beta\,X(t)G(t) + R_a(t)$$
$$\mathrm{d}X(t)/\mathrm{d}t = -\mu\,X(t) + \mu\,(I(t) - I_b)$$

to infer parameters and quantify input uncertainty given noisy measurement $G^{\eta}_{\mathrm{ref}}(t)$ and uncertain input data $R_a(t)$.



Using evaluation of the integral point-of-view: see, e.g., Dick, Gantner, Le Gia, Nuyens, Scheichl, Schillings, Schwab, Stuart, Teckentrup, . . .

Example: Helmholtz equation ($s = \infty$)

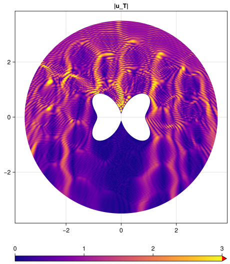
Exterior Dirichlet problem on $\mathbb{R}^d \setminus S$ such that $u(\boldsymbol{x}, \boldsymbol{y})$ satisfies

$$-\nabla \cdot (A(\boldsymbol{x}, \boldsymbol{y}) \nabla u(\boldsymbol{x}, \boldsymbol{y}) - k^2 \, n(\boldsymbol{x}, \boldsymbol{y}) \, u(\boldsymbol{x}, \boldsymbol{y}) = -f(\boldsymbol{x}) \qquad \text{outside of } S,$$

with $u = 0$ on ∂S and the Sommerfeld radiation condition

$$\partial_r u(\boldsymbol{x}, \boldsymbol{y}) - \mathrm{i}\, k \, u(\boldsymbol{x}, \boldsymbol{y}) \in o(r^{-(d-1)/2}) \text{ for } r = \|x\|_2 \to \infty.$$

Example QoI = expected far field pattern for random field $n(\boldsymbol{x}, \boldsymbol{y})$.



Graham, Kuo, Nuyens, Spence, Sloan (in preparation)

Example: DNN for PDE / function approximation (s large, ∞)

Assuming Chebyshev basis for random field (where $y_j \in [-1,1]$ e.g. Adcock, Brugiapaglia, Webster (2022)) but formulated in terms of uniform $y_j \in [0,1]$ variables:

$$a(\boldsymbol{x},\boldsymbol{y}) = \phi_0(\boldsymbol{x}) + \sum_{j\ge1} \sin(2\pi y_j)\,\phi_j(\boldsymbol{x}).$$

Use n training examples $(\boldsymbol{y}, G)$ to optimize DNN. Training error is used as proxy for real error / generalization error / L_2 approximation error:

$$E_T^2(\theta) = \frac{1}{n}\sum_{k=1}^{n} |G(\boldsymbol{y}^{(k)}) - G_\theta(\boldsymbol{y}^{(k)})|^2,$$

$$E_G^2(\theta) = \int_{[0,1]^s} |G(\boldsymbol{y}) - G_\theta(\boldsymbol{y})|^2 \,\mathrm{d}\boldsymbol{y} = \|G - G_\theta\|_{L_2}^2.$$

POV: You are approximating the integral of $|G - G_\theta|^2 \ldots$

Keller, Kuo, Nuyens, Sloan (in preparation); Mishra, Rusch (2021), Longo, Mishra, Rusch, Schwab (2021), . . .

10. Obtaining error bounds for the diffusion example

Error analysis for QMC part

Calculate expected value of $G(u(\boldsymbol{x};\boldsymbol{y}))$ over $\boldsymbol{y} \sim \otimes_{j\geq 1}\mu_j(\mathbb{U})$

$$\mathbb{E}[G(u)] = \int_{\mathbb{U}^{\mathbb{N}}} G(u(\cdot;\boldsymbol{y}))\,\mu(\mathrm{d}\boldsymbol{y})$$

with $u(\cdot;\boldsymbol{y})$ the solution of the PDE for the parameter choice $\boldsymbol{y}$.

- This is an integral of a function $F(\boldsymbol{y}) = G(u(\cdot;\boldsymbol{y}))$. What with $\boldsymbol{x}$?
- For QMC convergence analysis: need to know what function space F belongs to. Easiest if G is a linear function that removes the effect of $\boldsymbol{x}$.

Demonstratation for "uniform case" $a = a_0 + \sum_{j\geq 1} y_j\varphi_j$ and for first order mixed derivatives (1st order convergence for QMC) and G a linear functional.

Solving the PDE: transform to weak form

Strong form of PDE demands $u \in V$ to satisfy, for given $\boldsymbol{y}$,

$$-\nabla \cdot a(\boldsymbol{x};\boldsymbol{y})\nabla u(\boldsymbol{x};\boldsymbol{y}) = f(\boldsymbol{x}).$$

The weak form demands $u \in V$ to satisfy, for given $\boldsymbol{y}$,

$$\int_\Omega -\nabla \cdot a(\boldsymbol{x};\boldsymbol{y})\nabla u(\boldsymbol{x};\boldsymbol{y})\, v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_\Omega f(\boldsymbol{x})\, v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \qquad \forall v \in V_0.$$

Solving the PDE: transform to weak form

Strong form of PDE demands $u \in V$ to satisfy, for given $\boldsymbol{y}$,

$$-\nabla \cdot a(\boldsymbol{x};\boldsymbol{y})\nabla u(\boldsymbol{x};\boldsymbol{y}) = f(\boldsymbol{x}).$$

The weak form demands $u \in V$ to satisfy, for given $\boldsymbol{y}$,

$$\int_\Omega a(\boldsymbol{x};\boldsymbol{y})\,\nabla u(\boldsymbol{x};\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_\Omega f(\boldsymbol{x})\,v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \qquad \forall v \in V_0.$$

For QMC analysis we have norms depending on mixed derivatives.
Want to know the sensitivity w.r.t. the parameters $(y_1, y_2, \ldots)$:

$$\partial_{y_j} u(\boldsymbol{x};\boldsymbol{y}) := \frac{\partial}{\partial y_j} u(\boldsymbol{x};\boldsymbol{y}), \quad \partial_{\boldsymbol{y}}^{\mathfrak{u}} u(\boldsymbol{x};\boldsymbol{y}) := \prod_{j\in\mathfrak{u}} \frac{\partial}{\partial y_j} u(\boldsymbol{x};\boldsymbol{y}) = \frac{\partial^{|\mathfrak{u}|}}{\partial \boldsymbol{y}_{\mathfrak{u}}} u(\boldsymbol{x};\boldsymbol{y}),$$

$$\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\boldsymbol{x};\boldsymbol{y}) := \prod_{j\in\mathfrak{u}} \frac{\partial^{\tau_j}}{\partial y_j^{\tau_j}} u(\boldsymbol{x};\boldsymbol{y}) = \frac{\partial^{|\boldsymbol{\tau}|}}{\partial \boldsymbol{y}^{\boldsymbol{\tau}}} u(\boldsymbol{x};\boldsymbol{y}).$$

Towards the norm for the QMC cubature

For QMC we want to bound a norm like

$$\|F\|_{\alpha}^{2} := \sum_{\boldsymbol{\tau} \in \{0:\alpha\}^{s}} \gamma_{\mathrm{supp}(\boldsymbol{\tau})}^{-2} \|\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} F\|_{L_2}^{2}.$$

But our function $F = G(u(\boldsymbol{x};\boldsymbol{y}))$.
Assume G is a linear functional then for any $\boldsymbol{y}$

$$\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} G(u(\boldsymbol{x};\boldsymbol{y}))(\boldsymbol{y}) = G(\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\boldsymbol{x};\boldsymbol{y})) \leq \|G\|_{V^*} \|\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\cdot;\boldsymbol{y})\|_{V}.$$

Define a corresponding Bochner norm

$$\|u\|_{\alpha,\mathbb{U}^s,V}^{2} := \sum_{\boldsymbol{\tau} \in \{0:\alpha\}^{s}} \gamma_{\mathrm{supp}(\boldsymbol{\tau})}^{-2} \|\|\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\cdot;\boldsymbol{y})\|_{V}^{2}\|_{L_2}^{2}.$$

We have

$$\|G(u(\cdot;\boldsymbol{y}))\|_{\alpha,\mathbb{U}^s,V} \leq \|G\|_{V^*} \|u(\cdot;\boldsymbol{y})\|_{\alpha,\mathbb{U}^s,V}.$$

To bound $\|u\|_{\alpha,\mathbb{U}^s,V}$ we need bounds on $\|\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\cdot;\boldsymbol{y})\|_{V}$ for given $\boldsymbol{y}$.

Differentiate under the integral in the weak form

Apply derivative to weak form, for all $v \in V_0$ and for any given $\boldsymbol{y}$:

$$\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} \int_\Omega a(\boldsymbol{x};\boldsymbol{y})\,\nabla u(\boldsymbol{x};\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} \int_\Omega f(\boldsymbol{x})\,v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

$$\Leftrightarrow \int_\Omega \partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} \left(a(\boldsymbol{x};\boldsymbol{y})\,\nabla u(\boldsymbol{x};\boldsymbol{y})\right) \cdot \nabla v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = 0.$$

Use Leibniz formula:

$$\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} \left(a(\boldsymbol{x};\boldsymbol{y})\,\nabla u(\boldsymbol{x};\boldsymbol{y})\right) = \sum_{\boldsymbol{\omega} \le \boldsymbol{\tau}} \binom{\boldsymbol{\tau}}{\boldsymbol{\omega}} \partial_{\boldsymbol{y}}^{\boldsymbol{\omega}} \left(a(\boldsymbol{x};\boldsymbol{y})\right) \partial_{\boldsymbol{y}}^{\boldsymbol{\tau}-\boldsymbol{\omega}} \left(\nabla u(\boldsymbol{x};\boldsymbol{y})\right)$$

"uniform" case, $a(\boldsymbol{x};\boldsymbol{y}) = a_0(\boldsymbol{x}) + \sum_{j \ge 1} y_j\,\varphi_j(\boldsymbol{x})$:

$$= \sum_{\substack{\boldsymbol{\omega} \le \boldsymbol{\tau} \\ |\boldsymbol{\omega}| \le 1}} \binom{\boldsymbol{\tau}}{\boldsymbol{\omega}} \partial_{\boldsymbol{y}}^{\boldsymbol{\omega}} \left(a(\boldsymbol{x};\boldsymbol{y})\right) \partial_{\boldsymbol{y}}^{\boldsymbol{\tau}-\boldsymbol{\omega}} \left(\nabla u(\boldsymbol{x};\boldsymbol{y})\right)$$

$$= a(\boldsymbol{x};\boldsymbol{y})\,\nabla \partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\boldsymbol{x};\boldsymbol{y}) + \sum_{j \in \mathrm{supp}(\boldsymbol{\tau})} \tau_j\,\varphi_j(\boldsymbol{x})\,\nabla \partial_{\boldsymbol{y}}^{\boldsymbol{\tau}-\boldsymbol{e}_j} u(\boldsymbol{x};\boldsymbol{y}).$$

Continued…

Thus for any given $\boldsymbol{y}$ and for any $v \in V_0$ the solution u satisfies

$$\int_\Omega a(\boldsymbol{x};\boldsymbol{y})\,\nabla\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\boldsymbol{x};\boldsymbol{y})\cdot\nabla v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = -\sum_{j\in\mathrm{supp}(\boldsymbol{\tau})}\tau_j\int_\Omega \varphi_j(\boldsymbol{x})\,\nabla\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}-\boldsymbol{e}_j} u(\boldsymbol{x};\boldsymbol{y})\cdot\nabla v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}.$$

We are free to choose $v(\boldsymbol{x}) = \partial^{\boldsymbol{\tau}} u(\boldsymbol{x};\boldsymbol{y})$. Define the energy norm (Cohen, Bachmayr, Migliorati)

$$\|\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\cdot;\boldsymbol{y})\|_{a_{\boldsymbol{y}}}^2 := \int_\Omega a(\boldsymbol{x};\boldsymbol{y})\,|\nabla\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\boldsymbol{x};\boldsymbol{y})|^2\,\mathrm{d}\boldsymbol{x}$$

and the V-norm

$$\|\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\cdot;\boldsymbol{y})\|_V^2 := \int_\Omega |\nabla\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\boldsymbol{x};\boldsymbol{y})|^2\,\mathrm{d}\boldsymbol{x} \le a_{\min}^{-1}\,\|\partial^{\boldsymbol{\tau}} u(\cdot;\boldsymbol{y})\|_{a_{\boldsymbol{y}}}^2.$$

(Note that for the solution of the PDE we need $a(\boldsymbol{x};\boldsymbol{y}) \ge a_{\min} > 0$.)

Continued… (2)

Thus we have

$$\|\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\cdot;\boldsymbol{y})\|_{a_{\boldsymbol{y}}}^2 = -\int_\Omega \sum_{j\in\mathrm{supp}(\boldsymbol{\tau})} \tau_j\,\varphi_j(\boldsymbol{x})\,\nabla\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}-\boldsymbol{e}_j} u(\boldsymbol{x};\boldsymbol{y})\cdot\nabla\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\boldsymbol{x};\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}$$

Now there are different routes in different papers. We follow a particularly nice one (imho). (Refs: Kazashi, Schwab & Herrmann) For some $b_j > 0$ and $\boldsymbol{b}^{\boldsymbol{\tau}} = \prod_{j\geq 1} b_j^{\tau_j}$, apply Cauchy–Schwarz to

$$\left(\boldsymbol{b}^{-2\boldsymbol{\tau}} \sum_{j\in\mathrm{supp}(\boldsymbol{\tau})} \tau_j\,\varphi_j(\boldsymbol{x})\,\nabla\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}-\boldsymbol{e}_j} u(\boldsymbol{x};\boldsymbol{y})\cdot\nabla\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\boldsymbol{x};\boldsymbol{y})\right)^2$$

$$\leq \sum_{j\in\mathrm{supp}(\boldsymbol{\tau})} \boldsymbol{b}^{-\boldsymbol{e}_j}\,\tau_j\,|\varphi_j(\boldsymbol{x})|\,\left|\boldsymbol{b}^{-(\boldsymbol{\tau}-\boldsymbol{e}_j)}\,\nabla\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}-\boldsymbol{e}_j} u(\boldsymbol{x};\boldsymbol{y})\right|^2$$

$$\times \sum_{j\in\mathrm{supp}(\boldsymbol{\tau})} \boldsymbol{b}^{-\boldsymbol{e}_j}\,\tau_j\,|\varphi_j(\boldsymbol{x})|\,\left|\boldsymbol{b}^{-\boldsymbol{\tau}}\,\nabla\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\boldsymbol{x};\boldsymbol{y})\right|^2$$

Continued. . . (3)

Now we apply the next nice trick. (Refs: Cohen, DeVore, Schwab)
For some k consider the sum

$$\sum_{|\boldsymbol{\tau}|=k} \sum_{j\in\mathrm{supp}(\boldsymbol{\tau})} A_j(\tau_j)\, B_{\boldsymbol{\tau}-\boldsymbol{e}_j} = \sum_{|\boldsymbol{\omega}|=k-1} \sum_{j\geq 1} A_j(\omega_j+1)\, B_{\boldsymbol{\omega}}.$$

For our case, and limiting to 1st order, thus $\boldsymbol{\tau} \in \{0,1\}^s$, then

$$\sum_{\substack{\boldsymbol{\tau}\in\{0,1\}^s \\ |\boldsymbol{\tau}|=k}} \sum_{j\in\mathrm{supp}(\boldsymbol{\tau})} \tau_j \frac{|\varphi_j(\boldsymbol{x})|}{b_j} \left|\boldsymbol{b}^{-(\boldsymbol{\tau}-\boldsymbol{e}_j)} \nabla \partial_{\boldsymbol{y}}^{\boldsymbol{\tau}-\boldsymbol{e}_j} u(\boldsymbol{x};\boldsymbol{y})\right|^2$$

$$= \sum_{\substack{\boldsymbol{\omega}\in\{0,1\}^s \\ |\boldsymbol{\omega}|=k-1}} \sum_{j=1}^{s} \frac{|\varphi_j(\boldsymbol{x})|}{b_j} \left|\boldsymbol{b}^{-\boldsymbol{\omega}} \nabla \partial_{\boldsymbol{y}}^{\boldsymbol{\omega}} u(\boldsymbol{x};\boldsymbol{y})\right|^2.$$

We can use this to "reduced the order" from k to $k-1$.

Continued. . . (4)

Thus combining these tricks we obtain

$$\left(\sum_{\substack{\boldsymbol{\tau}\in\{0,1\}^s \\ |\boldsymbol{\tau}|=k}} \boldsymbol{b}^{-2\boldsymbol{\tau}} \, \|\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\cdot;\boldsymbol{y})\|_{a_{\boldsymbol{y}}}^2 \right)^2$$

$$\leq \int_\Omega \sum_{\substack{\boldsymbol{\omega}\in\{0,1\}^s \\ |\boldsymbol{\omega}|=k-1}} \sum_{j=1}^{s} \frac{|\varphi_j(\boldsymbol{x})|}{b_j} \left| \boldsymbol{b}^{-\boldsymbol{\omega}} \, \nabla \partial_{\boldsymbol{y}}^{\boldsymbol{\omega}} u(\boldsymbol{x};\boldsymbol{y}) \right|^2 \, \mathrm{d}\boldsymbol{x}$$

$$\times \int_\Omega \sum_{\substack{\boldsymbol{\tau}\in\{0,1\}^s \\ |\boldsymbol{\tau}|=k}} \sum_{j\in\mathrm{supp}(\boldsymbol{\tau})} \frac{|\varphi_j(\boldsymbol{x})|}{b_j} \left| \boldsymbol{b}^{-\boldsymbol{\tau}} \, \nabla \partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\boldsymbol{x};\boldsymbol{y}) \right|^2 \, \mathrm{d}\boldsymbol{x}$$

Continued. . . (5)

Rewrite:

$$\left(\sum_{\substack{\boldsymbol{\tau}\in\{0,1\}^s \\ |\boldsymbol{\tau}|=k}} \boldsymbol{b}^{-2\boldsymbol{\tau}} \, \|\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\cdot;\boldsymbol{y})\|_{a_{\boldsymbol{y}}}^2 \right)^2$$

$$\leq \left(\sum_{j=1}^{s} \frac{\|\varphi_j\|_{L_\infty}}{b_j \, a_{\min}} \right)^2 \sum_{\substack{\boldsymbol{\omega}\in\{0,1\}^s \\ |\boldsymbol{\omega}|=k-1}} \boldsymbol{b}^{-2\boldsymbol{\omega}} \|\partial_{\boldsymbol{y}}^{\boldsymbol{\omega}} u(\cdot;\boldsymbol{y})\|_{a_{\boldsymbol{y}}}^2$$

$$\times \sum_{\substack{\boldsymbol{\tau}\in\{0,1\}^s \\ |\boldsymbol{\tau}|=k}} \boldsymbol{b}^{-2\boldsymbol{\tau}} \|\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\cdot;\boldsymbol{y})\|_{a_{\boldsymbol{y}}}^2$$

Continued. . . (6)

Finally, enclosing all with the trick from Kuo, Schwab, Sloan (2012),

$$\sum_{\substack{\boldsymbol{\tau}\in\{0,1\}^s\\|\boldsymbol{\tau}|=k}} \boldsymbol{b}^{-2\boldsymbol{\tau}} \|\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\cdot;\boldsymbol{y})\|_{a_{\boldsymbol{y}}}^2$$

$$\leq \left(\sum_{j=1}^{s} \frac{\|\varphi_j\|_{L_\infty}}{b_j\, a_{\min}}\right)^2 \sum_{\substack{\boldsymbol{\omega}\in\{0,1\}^s\\|\boldsymbol{\omega}|=k-1}} \boldsymbol{b}^{-2\boldsymbol{\omega}} \|\partial_{\boldsymbol{y}}^{\boldsymbol{\omega}} u(\cdot;\boldsymbol{y})\|_{a_{\boldsymbol{y}}}^2$$

$$\leq \left(\sum_{j=1}^{s} \frac{\|\varphi_j\|_{L_\infty}}{b_j\, a_{\min}}\right)^{2k} \|u(\cdot;\boldsymbol{y})\|_{a_{\boldsymbol{y}}}^2,$$

and

$$\sum_{\substack{\boldsymbol{\tau}\in\{0,1\}^s\\|\boldsymbol{\tau}|=k}} \boldsymbol{b}^{-2\boldsymbol{\tau}} \|\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\cdot;\boldsymbol{y})\|_V^2 \leq \frac{a_{\max}}{a_{\min}} \left(\sum_{j=1}^{s} \frac{\|\varphi_j\|_{L_\infty}}{b_j\, a_{\min}}\right)^{2k} \|u(\cdot;\boldsymbol{y})\|_V^2.$$

Continued. . . (7)

Now we are finally where we need to be.
From the last expression we obtain

$$\boldsymbol{b}^{-2\boldsymbol{\tau}} \, \|\partial_{\boldsymbol{y}}^{\boldsymbol{\tau}} u(\cdot;\boldsymbol{y})\|_V^2 \le \frac{a_{\max}}{a_{\min}} \left(\sum_{j=1}^{s} \frac{\|\varphi_j\|_{L_\infty}}{b_j \, a_{\min}} \right)^{2\,|\boldsymbol{\tau}|} \quad \|u(\cdot;\boldsymbol{y})\|_V^2$$

Further, from the PDE we know the a priori bound

$$\|u(\cdot;\boldsymbol{y})\|_V \le \frac{\|f\|_{V^*}}{a_{\min}},$$

note that for the "uniform case" this bound is uniform in $\boldsymbol{y}$.

We ended up with product weights.
The bound is independent of s if we can find a sequence b_j such that

$$\sum_{j\ge1} \frac{\|\varphi_j\|_{L_\infty}}{b_j \, a_{\min}} < 1.$$

Modern view of QMC error bounds

For the QMC error we end up with bounds for $F(\boldsymbol{y}) = G(u_h^s(\boldsymbol{x}, \boldsymbol{y}))$ of the form

$$|I(F) - Q_n(F)| \le \|F\|_{s,\alpha,\boldsymbol{\gamma}} \, \mathrm{wce}_{s,\alpha,\boldsymbol{\gamma}}(n)$$

which basically looks like this:

$$\Big(\sum_{\mathfrak{u} \subseteq \{1,\ldots,s\}} \gamma_{\mathfrak{u}}^{-1} \, A_{\mathfrak{u}} \Big)^{1/2} \Big(\sum_{\mathfrak{u} \subseteq \{1,\ldots,s\}} \gamma_{\mathfrak{u}}^{\lambda} \, B_{\mathfrak{u}} \Big)^{1/2\lambda}.$$

Modern view, e.g., Kuo, Sloan, Schwab (2012), is to minimize the upper bound by choosing

$$\gamma_{\mathfrak{u}} = \left(\frac{A_{\mathfrak{u}}}{B_{\mathfrak{u}}} \right)^{1/(1+\lambda)}.$$

Nguyen, Nuyens (2021a, 2021b).

11. Numerical example of porous flow

Example of porous flow with circulant embedding

Assume the following elliptic PDE

$$-\nabla \cdot \big(a(\boldsymbol{x},\omega)\nabla u(\boldsymbol{x},\omega)\big) = f(\boldsymbol{x}), \qquad \text{for } \boldsymbol{x} \in D,\ \text{a.s. } \omega \in \Omega,$$

and $u(\boldsymbol{x},\omega) = 0$ for $\boldsymbol{x} \in \delta D$, where

Example of porous flow with circulant embedding

Assume the following elliptic PDE

$$-\nabla \cdot \big(a(\boldsymbol{x},\omega)\nabla u(\boldsymbol{x},\omega)\big) = f(\boldsymbol{x}), \qquad \text{for } \boldsymbol{x} \in D, \text{ a.s. } \omega \in \Omega,$$

and $u(\boldsymbol{x},\omega) = 0$ for $\boldsymbol{x} \in \delta D$, where

- $D \subset \mathbb{R}^d$ is a "nice" bounded physical domain, $d = 1, 2, 3$,

Example of porous flow with circulant embedding

Assume the following elliptic PDE

$$-\nabla \cdot \big(a(\boldsymbol{x},\omega)\nabla u(\boldsymbol{x},\omega)\big) = f(\boldsymbol{x}), \qquad \text{for } \boldsymbol{x} \in D,\ \text{a.s. } \omega \in \Omega,$$

and $u(\boldsymbol{x},\omega) = 0$ for $\boldsymbol{x} \in \delta D$, where

- $D \subset \mathbb{R}^d$ is a "nice" bounded physical domain, $d = 1, 2, 3$,
- ω is a random event from $(\Omega, \mathcal{A}, \mathbb{P})$,

Example of porous flow with circulant embedding

Assume the following elliptic PDE

$$-\nabla \cdot \big(a(\boldsymbol{x},\omega)\nabla u(\boldsymbol{x},\omega)\big) = f(\boldsymbol{x}), \qquad \text{for } \boldsymbol{x} \in D,\ \text{a.s. } \omega \in \Omega,$$

and $u(\boldsymbol{x},\omega) = 0$ for $\boldsymbol{x} \in \delta D$, where

- $D \subset \mathbb{R}^d$ is a "nice" bounded physical domain, $d = 1, 2, 3$,
- ω is a random event from $(\Omega, \mathcal{A}, \mathbb{P})$,
- $a(\boldsymbol{x},\omega)$ is a scalar lognormal random field:

$$a(\boldsymbol{x},\omega) = \exp(Z(\boldsymbol{x},\omega))$$

Example of porous flow with circulant embedding

Assume the following elliptic PDE

$$-\nabla \cdot \big(a(\boldsymbol{x},\omega)\nabla u(\boldsymbol{x},\omega)\big) = f(\boldsymbol{x}), \qquad \text{for } \boldsymbol{x} \in D, \text{ a.s. } \omega \in \Omega,$$

and $u(\boldsymbol{x},\omega) = 0$ for $\boldsymbol{x} \in \delta D$, where

- $D \subset \mathbb{R}^d$ is a "nice" bounded physical domain, $d = 1, 2, 3$,
- ω is a random event from $(\Omega, \mathcal{A}, \mathbb{P})$,
- $a(\boldsymbol{x},\omega)$ is a scalar lognormal random field:

$$a(\boldsymbol{x},\omega) = \exp(Z(\boldsymbol{x},\omega))$$

with $Z(\boldsymbol{x},\omega)$ a homogenous Gaussian random field with Matérn covariance function with parameter $\nu > 1/2$

$$r(\boldsymbol{x},\boldsymbol{x}') = \rho_\nu(r) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} (\alpha r)^\nu K_\nu(\alpha r), \qquad \alpha = 2\sqrt{\nu}/\lambda,$$

with $r = \|\boldsymbol{x} - \boldsymbol{x}'\|_p$, λ the length scale and σ^2 the variance.

From random field to parametrised problem

We parametrise the random event ω by a vector $\boldsymbol{y}(\omega) \in \mathbb{R}^\infty$.

E.g., using the Karhunen-Loève (KL) expansion:

$$a(\boldsymbol{x}, \omega) \equiv a(\boldsymbol{x}, \boldsymbol{y}) = a_0(\boldsymbol{x}) \ \exp\left(\sum_{j\geq 1} y_j \sqrt{\mu_j}\, \xi_j(\boldsymbol{x})\right),$$

with $y_j \sim N(0,1)$, and $\{(\mu_j, \xi_j)\}_{j\geq 1}$ is the sequence of ordered eigenvalues and eigenfunctions of the integral operator

$$(\mathcal{R}\,\xi)(\boldsymbol{x}) = \int_D \rho_\nu(\|\boldsymbol{x} - \boldsymbol{x}'\|_p)\, \xi(\boldsymbol{x}')\, \mathrm{d}\boldsymbol{x}'.$$

From random field to parametrised problem

We parametrise the random event ω by a vector $\boldsymbol{y}(\omega) \in \mathbb{R}^\infty$.

E.g., using the Karhunen-Loève (KL) expansion:

$$a(\boldsymbol{x},\omega) \equiv a(\boldsymbol{x},\boldsymbol{y}) = a_0(\boldsymbol{x})\ \exp\left(\sum_{j\geq 1} y_j \sqrt{\mu_j}\,\xi_j(\boldsymbol{x})\right),$$

with $y_j \sim N(0,1)$, and $\{(\mu_j,\xi_j)\}_{j\geq 1}$ is the sequence of ordered eigenvalues and eigenfunctions of the integral operator

$$(\mathcal{R}\,\xi)(\boldsymbol{x}) = \int_D \rho_\nu(\|\boldsymbol{x}-\boldsymbol{x}'\|_p)\,\xi(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}'.$$

The "roughness" depends on the summability of $b_j = \sqrt{\mu_j}\,\|\xi_j\|_{L^\infty}$, for $0 < p < 1$ (smaller is smoother) [Cohen, DeVore, Schwab'10]:

$$\sum_{j\geq 1}\left(\sqrt{\mu_j}\,\|\xi_j\|_{L^\infty}\right)^p = \sum_{j\geq 1} b_j^p < \infty.$$

Aim: Calculate expected value of quantity of interest

Given ω and corresponding solution $u(\boldsymbol{x},\omega)$ we are interested in

$$\begin{aligned}\mathbb{E}_\omega\left[\mathcal{G}(u(\boldsymbol{x},\omega))\right] &= \int_{\mathbb{R}^\infty} \mathcal{G}(u(\boldsymbol{x},\boldsymbol{y}))\,\phi(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \\ &= \int_{(0,1)^\infty} \mathcal{G}(u(\boldsymbol{x},\boldsymbol{\Phi}^{-1}(\boldsymbol{y})))\,\mathrm{d}\boldsymbol{y},\end{aligned}$$

for some linear functional $\mathcal{G}$ acting on the physical space $\boldsymbol{x}$, with ϕ the multivariate normal and $\boldsymbol{\Phi}^{-1}$ it's cumulative inverse.

Aim: Calculate expected value of quantity of interest

Given ω and corresponding solution $u(\boldsymbol{x},\omega)$ we are interested in

$$\mathbb{E}_\omega\left[\mathcal{G}(u(\boldsymbol{x},\omega))\right] = \int_{\mathbb{R}^\infty} \mathcal{G}(u(\boldsymbol{x},\boldsymbol{y}))\,\phi(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}$$
$$= \int_{(0,1)^\infty} \mathcal{G}(u(\boldsymbol{x},\boldsymbol{\Phi}^{-1}(\boldsymbol{y})))\,\mathrm{d}\boldsymbol{y},$$

for some linear functional $\mathcal{G}$ acting on the physical space $\boldsymbol{x}$, with ϕ the multivariate normal and $\boldsymbol{\Phi}^{-1}$ it's cumulative inverse.

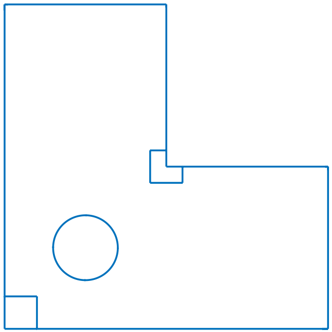
The error breaks down in three pieces:

1. $u(\boldsymbol{x},\boldsymbol{y}) \approx u_h(\boldsymbol{x},\boldsymbol{y})$ (FEM discretisation + quadrature),
2. $\int_{(0,1)^\infty} \approx \int_{(0,1)^s}$ (KL-trunc. or avoid by circ. emb. $\rightarrow$ intp. err.)
3. $\int_{(0,1)^s} \approx \frac{1}{N}\sum_{k=1}^N$ (quadrature approximation for $\mathbb{E}_\omega$).

Circulant embedding e.g. [Dietrich,Newsam'97;Graham,Kuo,Nuyens,Scheichl,Sloan'11].

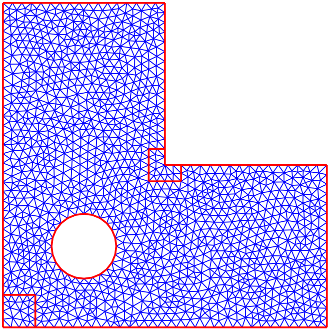
A 2D discretisation example

1. Discretise using linear elements to represent u_h.
2. Approximate Galerkin integrals with midpoint rule on each element τ (single evaluation in center of mass $\boldsymbol{x}_\tau^C$).
3. Approximate $a(\boldsymbol{x}_\tau^C, \omega)$ by multilinear interpolation on the regular grid.



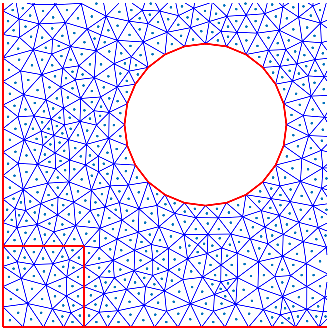
A 2D discretisation example

1. Discretise using linear elements to represent u_h.
2. Approximate Galerkin integrals with midpoint rule on each element τ (single evaluation in center of mass $\boldsymbol{x}_\tau^C$).
3. Approximate $a(\boldsymbol{x}_\tau^C, \omega)$ by multilinear interpolation on the regular grid.



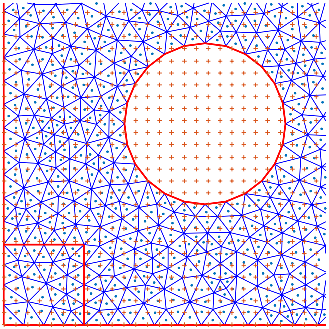
A 2D discretisation example

1. Discretise using linear elements to represent u_h.
2. Approximate Galerkin integrals with midpoint rule on each element τ (single evaluation in center of mass $\boldsymbol{x}_\tau^C$).
3. Approximate $a(\boldsymbol{x}_\tau^C, \omega)$ by multilinear interpolation on the regular grid.



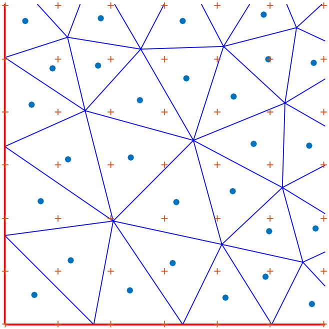
A 2D discretisation example

1. Discretise using linear elements to represent u_h.
2. Approximate Galerkin integrals with midpoint rule on each element τ (single evaluation in center of mass $\boldsymbol{x}_\tau^C$).
3. Approximate $a(\boldsymbol{x}_\tau^C, \omega)$ by multilinear interpolation on the regular grid.



A 2D discretisation example

1. Discretise using linear elements to represent u_h.
2. Approximate Galerkin integrals with midpoint rule on each element τ (single evaluation in center of mass $\boldsymbol{x}_\tau^C$).
3. Approximate $a(\boldsymbol{x}_\tau^C, \omega)$ by multilinear interpolation on the regular grid.



Dimension independent error estimate for $\mathbb{E}_\omega\left[\mathcal{G}(u(\boldsymbol{x},\omega))\right]$ using randomly shifted lattice rule

Theorem ([Graham,Kuo,Nuyens,Scheichl,Sloan '18b])

If $\|\boldsymbol{b}_{(s)}\|_p \le c$, *with* $1 \ge p > \frac{2}{3}$,

$$\|F\|_{s,\gamma} \lesssim \left(\sum_{\mathfrak{u}\subseteq\{1:s\}} \gamma_{\mathfrak{u}}^{-1} \left(\frac{|\mathfrak{u}|!}{(\log 2)^{|\mathfrak{u}|}} \right)^2 \prod_{j\in\mathfrak{u}} \frac{\tilde{b}_j^2}{\alpha_j - b_j} \right)^{1/2}$$

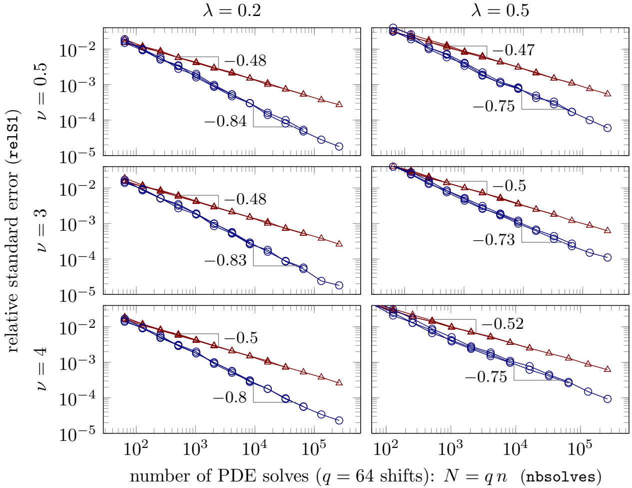
with all $\alpha_j > b_j$, $\tilde{b}_j = b_j/(2\exp(b_j^2/2)\Phi(b_j))$, *then*

$$RMS_{\boldsymbol{\Delta}}\left(I_s(F) - Q_{s,n}(F,\boldsymbol{\Delta})\right) \lesssim C_p\, n^{-\frac{1}{p}+\frac{1}{2}},$$

where $C_p \to \infty$ *for* $p \to \frac{2}{3}$.

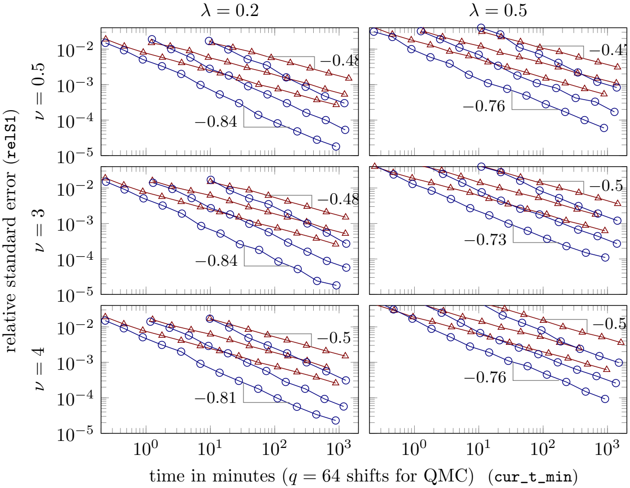
3D porous flow example

$m_0 = 7, 14, 28,\ h = 0.24, 0.12, 0.06,\ s = 2\,744, \ldots, 37\,933\,056$ stochastic dimensions



3D porous flow example

$m_0 = 7, 14, 28,\ h = 0.24, 0.12, 0.06,\ s = 2\,744, \ldots, 37\,933\,056$ stochastic dimensions



12. DNN regularity bounds

DNNs and function approximation

Elliptic PDEs with coefficients in Chebyshev / periodic setting.
$G(\boldsymbol{y})$ could e.g. be $u_{s,h}(\boldsymbol{x}^\dagger, \boldsymbol{y})$. Approximate G by a DNN.
This motivates the "periodic" DNN, $G_\theta = G_\theta^{[L]}$,

$$\begin{cases} G_\theta^{[0]}(\boldsymbol{y}) := W_0 \sin(2\pi \boldsymbol{y}) + \boldsymbol{b}_0 \\ G_\theta^{[\ell]}(\boldsymbol{y}) := W_\ell(\sigma(G_\theta^{[\ell-1]}(\boldsymbol{y}))) + \boldsymbol{b}_\ell, \quad \ell = 1, \ldots, L. \end{cases}$$

- W_ℓ is a $d_{\ell+1} \times d_\ell$ matrix
- $\boldsymbol{b}_\ell$ is a $d_{\ell+1} \times 1$ vector
- $d_0 = s$ is the dimension of the input vector $\boldsymbol{y}$
- $d_{L+1} = 1$ (for the example of point evaluation)
- σ is a smooth activation function

Keller, Kuo, Nuyens, Sloan (2024+)

Error analysis: approximation to integration

Mishra, Rusch (2021); Longo, Mishra, Rusch, Schwab (2021)

- Generalization error (L_2 approximation error)

$$E_G = E_G(\theta) := \left(\int_{[0,1]^s} |G(\boldsymbol{y}) - G_\theta(\boldsymbol{y})|^2 \, \mathrm{d}\boldsymbol{y} \right)^{1/2} = \|G - G_\theta\|_{L_2}$$

- Training error

$$E_T = E_T(\theta) := \left(\frac{1}{n} \sum_{k=0}^{n-1} |G(\boldsymbol{t}_k) - G_\theta(\boldsymbol{t}_k)|^2 \right)^{1/2} \quad \text{using lattice points}$$

- Generalization gap

$$|E_G - E_T| \le \sqrt{|E_G^2 - E_T^2|} \qquad \text{quadrature error for } (G - G_\theta)^2$$

- Hence

$$E_G \le E_T + |E_G - E_T| \le E_T + C_\Theta \, e^{\mathrm{int}}(Q_{s,n,\boldsymbol{z}}, \mathcal{H}_{s,\alpha,\boldsymbol{\gamma}}) \, \|(G - G_\theta)^2\|_{s,\alpha,\boldsymbol{\gamma}} \sim n^{-\alpha}$$

Keller, Kuo, Nuyens, Sloan (2024+)

Regularity periodic DNN for L_2 approximation

We have $G_\theta = G_\theta^{[L]}$ with

$$\begin{cases} G_\theta^{[0]}(\boldsymbol{y}) := W_0 \sin(2\pi \boldsymbol{y}) + \boldsymbol{b}_0 \\ G_\theta^{[\ell]}(\boldsymbol{y}) := W_\ell(\sigma(G_\theta^{[\ell-1]}(\boldsymbol{y}))) + \boldsymbol{b}_\ell, \quad \ell = 1, \ldots, L \end{cases}$$

Theorem. Keller, Kuo, Nuyens, Sloan (2024+)

$$|\partial^{\boldsymbol{\nu}} G_\theta(\boldsymbol{y})| \leq (2\pi\, R_1 \cdots R_{L-1})^{|\boldsymbol{\nu}|}\, R_L\, A_{L,|\boldsymbol{\nu}|} \prod_{j=1}^{s} T_{\nu_j}(\beta_j)$$

- $\max_{1 \leq k \leq d_1} |W_{0,k,j}| \leq \beta_j, \quad j = 1, \ldots, s$
- $\max_{1 \leq k \leq d_{\ell+1}} \sum_{1 \leq j \leq d_\ell} |W_{\ell,k,j}| \leq R_\ell, \quad \ell = 1, \ldots, L$
- $A_{L,n}$ depends on activation function

Hence the difference between approximation error and training error will converge like $n^{-\alpha}$ if you take training points on QMC point sets. (Periodic and non-periodic.) See also Longo, Mishra, Rusch, Schwab (2021) for higher-order polynomial lattice rules.

The end!

The end!

Thank you.